Using Personalization to Improve XML Retrieval

Luis M. de Campos, Juan M. Fernández-Luna, Juan F. Huete, and Eduardo Vicente-López

Abstract—As the amount of information increases every day and the users normally formulate short and ambiguous queries, personalized search techniques are becoming almost a must. Using the information about the user stored in a user profile, these techniques retrieve results that are closer to the user preferences. On the other hand, the information is being stored more and more in an semi-structured way, and XML has emerged as a standard for representing and exchanging this type of data. XML search allows a higher retrieval effectiveness, due to its ability to retrieve and to show the user specific parts of the documents instead of the full document. In this paper we propose several personalization techniques in the context of XML retrieval. We try to combine the different approaches where personalization may be applied: query reformulation, reranking of results and retrieval model modification. The experimental results obtained from a user study using a parliamentary document collection support the validity of our approach.

Index Terms—Information Retrieval, XML, Personalization, Query Expansion, Reranking, CAS queries.

1 Introduction

Over the last few years the amount of digital information has increased exponentially. Therefore, the use of Information Retrieval Systems (IRS) has become crucial in finding relevant information within this huge amount of data. These systems have been providing very good results for the majority of users. However, in addition to the aforementioned huge rise of digital information, there is the fact that users do not always specify accurately enough their information needs (they tend to formulate short and ambiguous queries). It is then inevitable that the access to the relevant information is becoming more difficult each day.

These aforementioned factors have led to a growing interest in personalization techniques [19], [44], [48]. In this context, personalization can be defined as the process by which, using information about the user, generally stored in a user profile, and the issued query, the most appropriate results are provided with respect to the user interests and preferences. In this way, personalization minimizes the information overload of users, making it possible to better satisfy their information needs. Thanks to this potentiality, personalization has become one of the key challenges and hot research areas in the information retrieval field [1], [4].

Another key aspect of this amount of digital information is the increasing use of different types of documents, whose textual content is organised around a well defined structure. XML (eXtensible Markup Language) has recently emerged as the document standard for representing and exchanging this type of semi-structured data. XML data is self-describing through content-oriented tags, which let computers interpret the meaning of the stored data. XML allows us to explicitly represent the internal structure of documents, which should be considered as aggregates of interrelated units, instead of atomic entities. Classical IR is not able to exploit this characteristic to carry out a more focused retrieval. In fact, the main XML-IR asset [29] is to take advantage of the document's internal structure, allowing one to retrieve both specific parts of the documents (we will call these parts Structural Units, SU) and complete documents. This will depend on the user's needs and the distribution of relevant material across the different parts of the XML document.

This new structural characteristic requires new designs and/or adaptations of the traditional IR techniques and evaluation metrics. They cannot simply be reused under this new approach, because of the dependency between XML document components. This document component dependency causes the following two main XML intrinsic difficulties [28]: (1) near-misses, which are document components that are structurally related to relevant components, such as a neighbouring paragraph or a container section; (2) overlap, which refers to the situation when the same text fragment is referenced multiple times, for example where a paragraph and its container section are both retrieved. Due to these dependencies, the development of retrieval (and also personalization) techniques over XML documents implicates some difficulties in terms of design and evaluation.

The main goal of this article[1] is to develop and evaluate new personalization strategies designed for XML documents, which is a relatively unexplored area. We have considered approaches to be used in the three different steps where personalization may be applied (and their combinations): before search (query reformulation, in our case, query expansion and transformation on content-and-structure queries), after search (reranking of results) and within the retrieval process (modification of

- *Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S.I. Informática y de Telecomunicación, CITIC-UGR, Universidad de Granada, 18071-Granada, Spain.*
  E-mail: {lci,jmfluna,jhg,evicente}@decsai.ugr.es

[1] This paper is an extended version of the conference paper [12].

Digital Object Indentifier 10.1109/TKDE.2013.75

the retrieval model). We focus on the effective use of the information provided by the user profile rather than on the construction of the profile itself. Our personalization techniques are mainly designed for document collections such as digital libraries or corpuses of big organizations, more than for the web due to its great structural heterogeneity. However, most of the proposed personalization techniques could also be applied to flat (non structured) documents with almost no changes.

Our main contribution is the proposal and evaluation of several new personalization techniques in the context of XML retrieval. Most of them include new personalization aspects, such as the use of two retrieved lists of results in the reranking process, a modification in the search engine, or even the use of ’content and structure’ queries for personalization purposes. Observing the obtained experimental results, we can conclude that all of them provide very good performance improvement over using no personalization. We suggest to use the proposed techniques, if possible, in this order: retrieval model modification, content and structure approach and the reranking approach.

The remainder of the article is organized in the following way. We first give, in Section 2, an overview of the different personalization strategies existing in the literature. Then, in Section 3, we show our proposed personalization approaches. Section 4 describes the experimental methodology. This includes a description of the document collection considered, how we have obtained the user profiles and the relevance assessments, our evaluation method, and the obtained results and our conclusions. Finally, we finish in Section 5 with some general conclusions and proposals for future work.

2 Related Work

In this section, we give a general overview of the different personalization techniques found in the specific literature. We should focus on XML personalization, since it is the field our work belongs to. However, we believe that it is useful to first give a broader view of personalization, not necessarily confined to XML documents. We do not wish to make an exhaustive analysis but merely to show the main types of existing personalization techniques and, for those more similar to our proposals, to present the main differences between them.

So, we will start with some ideas relative to the representation of users’ preferences by means of user profiles, and then review some of the existing works about general personalization methods. Next, focusing on XML, we will comment on the different ways of querying XML documents and then describe specific XML personalization methods.

2.1 User Profiles

The first thing to take into account to deal with personalization is to have a good representation of the user information (his/her interests and preferences), which is stored in a user profile. The more accurately this information represents the user, the better the retrieved results for this user.

An accurate representation of the user profile is very important in order to obtain good retrieval results, but there is another key component: how to use this information, i.e. how good the whole retrieval process is in order to exploit the information stored in the user profile. In this article we focus on the effective use of the information provided by the user profile (the personalization strategies) rather than on the construction of the profile itself. Anyway, some comments about building profiles are necessary.

There are many studies on how to build a good profile representation, whose process has two key points: (1) information sources and acquisition techniques, and (2) user profile representation and updating.

The first point is beyond the scope of our research. Considering the second point, there are two main user profile representations: a set of weighted keywords or terms [45] and rich semantic based structures, sometimes enhanced with the use of ontologies [43], [50]. The most common representation for user profiles is the first one, which we will use. The (weighted) keywords can be automatically extracted from documents, other kind of sources or directly provided by the user. In our experiments, the keywords will be extracted from the document collection being considered, either automatically or using expert assessments, as explained in Section 4.2. Each keyword has an associated numerical weight representing the importance of the term for the user.

2.2 General Personalization Methods

We will classify the different personalization techniques according to where they utilize the user profile information within the retrieval process [46].

Query reformulation. In this case, personalization is applied before searching. The most used technique is query expansion with the user profile terms (called query augmentation in [38]), which is an easy and powerful technique. For example, Shen et al. [42] select appropriate terms from related preceding queries and corresponding search results to expand the current query. Chirita et al. [17] generate the additional query terms by analysing user data at increasing granularity levels and using external thesauri.

Query expansion is a technique in itself [13], which can be used for personalization but also for other cases, as relevance and pseudo-relevance (blind) feedback[2]. In these cases, the expansion terms are extracted from either the documents judged as relevant by the user or from the first retrieved documents, instead of using a profile. In general, when using query expansion recall

[2] Relevance feedback can be viewed as a method of short-term personalization [18].

is improved but usually at the expense of precision. However, if we use a combined recall/precision measure query expansion results in better retrieval effectiveness, according to recent experimental studies [13]. Query expansion suffers the so-called query-drift problem [31], [51]. It confuses the user because the retrieved results may not contain the query terms the user was looking for in the original query. This is due to the change in the underlying *intent* between the original query and its expanded form. Its effect may be particularly serious when applying query expansion to personalization. The reason is that the number of terms in the profile may be high and these terms can be highly unrelated to the original query terms. The usual way of dealing with this problem, especially in feedback applications, is to emphasize the original query terms with respect to the expansion terms, for example giving less weights to the expansion terms. Our basic expansion method will also follow this approach, by applying a global normalization factor over the profile terms used for expansion.

Reranking of results. In this strategy personalization is applied after the search has been executed. It tries to improve precision by reranking the top results retrieved from the original query, taking into account the user profile information. There are plenty of studies following this strategy. For example, Sugiyama et al. [45] use a keyword-based user profile and rerank the results based on the similarity between each web page and the user profile. Chirita et al. [16] focus on reranking the web search output according to the cosine distance between each page and a set of Desktop terms describing user interests. In these cases, the original ranking is not taken into account. Teevan et al. [47] and Matthijs and Radlinski [33] also propose methods to incorporate the original rank within the final ranking. In most of the studies about personalization using reranking, only the original query is submitted to the search engine. The effort of comparing the retrieved results with the profile information, in order to rerank these results, is carried out outside the retrieval model implemented by the search engine (which is almost unavoidable in the case of web search).

In a context different from personalization, namely that of methods for fusion of retrieved lists, Meister et al. [34], [35] rerank a list retrieved in response to a query utilizing a second list. This list is retrieved by using a different retrieval method and/or query representation, and exploiting inter-document similarities between the lists, so as to improve precision in the very top ranks. Their methods can be used in the context of blind feedback-based automatic query expansion by reranking the list produced by blind feedback using the list retrieved in response to the original query [34]. Similarly, Zighelnic and Kurland [51] fuse the results retrieved in response to the original query and to its expanded form. This contributes to alleviate the query-drift problem. Our approach to personalization using reranking is close to these studies but used in the opposite way: we use the results from the expanded query to rerank the results obtained by the original query.

Retrieval model modification. There are not many articles which modify the search engine retrieval model in order to account for the user profile. Most of them are focused on link analysis. Haveliwala [22] computed a topic oriented PageRank, in which 16 PageRank vectors, biased on each of the main topics of the Open Directory, were initially calculated off-line. Then they were combined at run-time based on the similarity between the user query and each of the 16 topics. Jeh and Widom [26] were able to manage arbitrary topic vectors instead of a predefined set of topics. In order to generate topic oriented rankings, Nie et al. [36] distributed the PageRank of a page across the topics it contains. Chang et al. [14] personalized HITS instead of PageRank. Lastly, outside the field of link analysis, Teevan et al. [47] modified the probabilistic ranking function BM25 by weighting terms appearing in the user profile higher. We will also propose a modification of the search engine used in our experiments, in order to treat differently the terms appearing in the profile from those appearing in the original query.

2.3 Querying XML Documents

The most straightforward and effective querying method for non-structured document collections is the well-known keyword search. One of its key advantages is simplicity, since users only need to specify the keywords they are interested in. However, XML document collections have both content and structure, and may be queried by content, structure or both. In the terminology used within the Initiative for the Evaluation of XML Retrieval (INEX), keyword queries are known as content-only (CO) queries. Content-and-structure (CAS) queries are those containing both structure and content constraints. There are state-of-the art querying languages such as XQuery[3] (supported by XPath[4]) or NEXI [49], that allow us to retrieve XML documents based on content and structure. But they have two key disadvantages: (1) they are complex to learn to use and (2) the users must know the structure of the documents, which most of the time is not the case. These query languages are more suitable for expert users, letting them to specify these kinds of SUs that will much better satisfy their information needs, in opposition to the classic keyword search.

In this paper, we keep the simple keyword search query interface, although we exploit XML structure during the query processing, so that the retrieved results can be any kind of document components. We have decided to take this approach, because the users from the user study we carried out knew neither the structure of the underlying XML document collection, nor any of the complex querying languages. Perhaps for these

[3] http://www.w3.org/standards/techs/xquery
[4] http://www.w3.org/standards/techs/xpath

reasons, although there are many IRS able to deal with XML documents[5], often these systems can only process CO queries. Nevertheless, a general method used to convert some of these only CO-able systems into a fully structured IRS, which can process CAS queries, has recently been proposed by de Campos et al. [8]. This may be useful in our case because some of the proposed personalization strategies internally transform the original CO queries submitted by the users into CAS queries.

2.4 XML Personalization Methods

XML search personalization is not a very explored research area yet, and we have found very few studies dealing with this topic. Amer-Yahia et al. [2] developed their XML personalization system PIMENT. This is a system which enables query personalization by query rewriting and answer ranking. It is composed of a profile repository that stores user profiles, a query customizer that rewrites user queries based on user profiles and a ranking module to rank query answers. In PIMENT a user profile is a set of rules in the form (condition, action, conclusion). The condition and conclusion parts are XQuery Full Text[6], and action can be to add, remove or replace. Whenever a query matches a rule condition, it is rewritten accordingly. However, the generation of the rules in the user profile requires the user's active participation. Chernishev [15] takes PIMENT architecture as the base, adding a feedback module which tries to extract, from query history, the user's awareness of the documents' structure. The query history contains user queries, query results, and user responses (e.g. the set of chosen items or the user time to examine a particular item). The user knowledge about the structure of the documents is stored in the user repository, which will be used in the query rewriting process. As query rewriting it uses a mechanism based on a modified and well-known technique of query rewriting called relaxations. Amer-Yahia et al. [3] extended their previous work to a new framework called PIMENTO. With this approach, the user profile is a set of scoping and ordering rules (SRs and ORs, respectively). SRs allow for narrowing or broadening the scope of the query, while ORs are used to enforce ranking preferences by reranking the results of the previously SRs modified queries. SRs may be conflicting due to their order of application and ORs may be ambiguous, although the authors describe an algorithm to detect and resolve conflicting SRs and ambiguous ORs. They also define an OR-aware top-k pruning algorithm to guarantee an efficient query personalization process. Our approach for XML personalization is fairly different from these studies, as we use a keyword-based user profile to expand the query (which is a much more simple process than using a set of rules), together with reranking methods and modification of the retrieval model.

As we have already discussed, relevance and blind feedback techniques, although different, are related in several ways to personalization. Therefore, it is also interesting to briefly review existing work on relevance and blind feedback over XML documents. Within this area, Mass and Mandelbrod [32] propose a component ranking algorithm for XML retrieval and show how to apply known relevance feedback algorithms from traditional IR on top of it, to achieve relevance feedback for XML. Pan [37] proposes query expansion based on ontological similarities. A query is firstly expanded with the use of a global ontology. Then, after the first round of feedback from the user, a specific ontology is built from some parts of the global ontology and the query itself. This new ontology is then used for each round of query expansion and modified according to the user feedback. De Campos et al. [7], [9] propose probabilistic methods for reweighting and expanding both CO and CAS queries (adding terms extracted from relevant components instead of terms extracted from complete documents). Hsu et at. [24] devise a context-aware approach for searching XML to improve the effectiveness of keyword search on XML via query expansion. They find a set of XML path expressions that capture the contextual meaning of a keyword query based on pseudo-feedback. Paths in the contexts of the query are used to expand the original query.

Schenkel and Theobald [40], [41] present a formal framework to integrate different dimensions of feedback, beyond content based feedback, into XML retrieval. Concretely, they present methods that expand a CO query into a CAS query based on relevance feedback, by taking into account the structured dimension of XML. Further advances in this direction have been more recently proposed by Hlaoua et al. [23]. One of our proposals for personalization is also based on transforming the original CO query into a CAS query that incorporates the profile terms.

3 Personalization Strategies

In this section we are going to describe the different approaches considered to perform personalization on XML documents. More specifically, we have designed several personalization strategies based on: query expansion (addition of terms coming from the user profile to the original query); reranking (combination of the output of two queries – the original and the expanded queries); conversion of CO queries to CAS queries, making the most of the structure of the documents; and finally, modifying the retrieval model in order to natively differentiate original query terms from profile terms. These strategies are applied in the three typical scenarios where personalization is implemented: before and after search time, and changing the search engine.

These approaches will be experimentally compared in Section 4. One of the principles guiding our research is

[5] The series of INEX Workshop proceedings is an excellent source of information, see, e.g. [21], [20].

[6] http://www.w3.org/TR/xpath-full-text-10/

that we want most of the work to be carried out by the search engine of the IRS. We do not want to use expensive additional processes or calculations in order to integrate the user profile information (as in many of the reranking strategies mentioned in Section 2).

We shall assume that we have an XML IRS that, given a query, returns a list of results ordered by decreasing values of the Relevance Status Value (RSV) or retrieval score assigned by the IRS. Each result is an SU of an XML document in the collection. The list of results contains, at most, a fixed number of SUs (1500 in the experiments in Section 4) and follows the "Focused" INEX Task specification [27], i.e. overlapping has been eliminated.

3.1 Normalized Query Expansion

The first approach we are going to use is simply query expansion: concretely, we add to the original query the first k terms in the profile. The profile terms are ranked in descending order of importance, so that we select the k terms which are of greater importance. The number k of added terms is a parameter that should be adjusted. This is a very easy and efficient technique and only requires to perform a longer query. But its main drawback is the aforementioned query-drift problem. The expanded (original+profile) query could retrieve results closer to the user profile itself than to the original query (which represents his/her current information need). Moreover, as we are dealing with XML documents, the added profile terms could also provoke an increase in the size of the retrieved SUs, as a bigger SU probably is necessary to accommodate the increased number of query terms. Both problems will become more pronounced as more profile terms are added. On the other hand, adding too few terms may cause a poor representation of the true preferences of the user, so that some kind of trade-off becomes necessary.

To alleviate these problems, we propose the use of a global normalization factor applied to the weights of the profile terms, making their influence over the expanded query weaker. It is a kind of upper bound for the weights of the profile terms, in order to differentiate their importance with respect to the original query terms. More precisely, let $t_1, \ldots, t_m$ be the original terms in the query and $t_{m+1}, \ldots, t_{m+k}$ be the first k terms in the profile, whose weights, within the profile, are $w_{m+1}, \ldots, w_{m+k}$. Let $0 < p_0 \leq 1$ be the normalization factor. Then, the expanded query is a weighted query composed of the original query terms, with weights equal to 1 ($p_i = 1,\ i = 1, \ldots, m$) and the expanded profile terms with weights

$$p_i = p_0 * \frac{w_i}{\max_{m+1 \leq i \leq m+k} w_i},\ i = m+1, \ldots, m+k.$$

In this way, the added profile terms can receive at most a fraction p_0 of the maximum weight attached to the original query terms. The normalization factor p_0 is another parameter to be adjusted.

3.2 Reranking

Another obvious and simple approach to exploit the information in the profile would be to formulate two different queries: the original query and the profile query (where the query terms are only the first k terms in the profile). Then the obtained lists of results would be combined in some way. This approach may be seen as a particular kind of reranking but has a main drawback: the overlapping degree between the two lists of results would likely be very low (because the query terms and the profile terms may be quite unrelated), and therefore, their simple combination would be worthless [34]. This approach of combining or fusing two different lists is more useful within pseudo-relevance feedback techniques, where the query terms in the additional query come from the top documents retrieved by the original query. As the results of the original query are probably related to the original query topic, the new query terms selected from these top retrieved results are also likely to be related to the original query topic. Thus, a greater overlap is expected between the results of the original and the additional query, and their combination makes more sense. But this is not the case with personalization, where the original query terms may have nothing or almost nothing in common with the profile terms. Moreover, in personalization the two result lists should not be considered as being equally important. The original query, which contains the current user information need, should be more important than the profile one, which should be considered as a kind of *context*.

However, what we could do is to replace in the previous approach the profile query with the expanded query obtained by normalized query expansion, as explained in the previous section. Performing original and original+profile queries is one way to avoid the almost null overlap which would exist between the rankings of the original and profile queries. Doing so, the amount of overlap between both result lists is greater and, at the same time, their combination does not distort the original query as much as the combination of the original query and the profile query. This also helps to avoid the query-drift problem. Moreover, as the original query is considered more important, we will use the expanded query to rerank the original query results[7]. The basic idea is to reward SUs in the original query results that match with some SUs in the results of the expanded query.

In order to carry out this reranking, an important question is to decide when the retrieved elements in the two result lists match. In the case of flat documents there is no problem: two documents match if they are the same document. However, with XML documents there is the possibility that an SU in a list overlaps with a different SU in the other list. In XML retrieval, if an SU is relevant, then its container or descendant SUs are also relevant

[7] The opposite approach (i.e. reranking the results of the expanded query using the original query) has been considered within the field of pseudo-relevance feedback [34].

(at least relevant at some degree). Therefore, we say that there is a match between two SUs belonging to different result lists when one SU is the same, a container or a descendant of the other.

We have developed three variations of this reranking strategy (Fig. 1 shows an example of how they work). Let L_O and L_E be the lists containing the original query results and the expanded query results respectively.

- **Hard reranking (HRR)**: the reranked list, L_{HRR}, will contain the SUs in L_O but will be rearranged according to the relative ordering of the SUs in L_E that match them. The SUs in L_O that do not match with any SU in L_E will be placed at the end of L_{HRR} (in the same relative order they had in L_O). For example, in Fig. 1, as the SUs A, B and C from L_O also appear, in a different order, in L_E, then they also appear in L_{HRR} with this order. However, the SU D appears in L_O but not in L_E, so that it is placed at the end of L_{HRR}. This is a strict reranking, as L_{HRR} contains exactly the same SUs than L_O but with the order dictated by L_E. The order in L_O is not taken into account, except for the SUs that do not match.
- **Soft reranking (SRR)**: the lists L_O and L_E are first normalized by the RSV of its first result (the greatest RSV). For each match between both lists, the normalized RSV of the SU in L_E is added to the corresponding RSV of the SU in L_O that matches it. Then L_O, with the modified RSVs, is reordered to obtain the reranked list L_{SRR}. With this reranking strategy, L_{SRR} also contains exactly the same SUs as L_O, but the final ranking is an additive combination of the rankings in L_O and L_E.
- **Include reranking (IRR)**: it is similar to soft reranking. The only difference is that the SUs in L_E which have not matched with any SU in L_O are also included in L_O, with its corresponding RSV. Then, as in the previous case, L_O is reordered to obtain the reranked list L_{IRR}. In this case, L_{IRR} can include some SUs from L_E which were not present in L_O.

An important characteristic of our reranking strategy is that we do not need any complex calculations (involving access to the documents) in order to rerank the original query results. We only need to submit two queries to the search engine (the original and the expanded, adding the profile terms) and then rerank the results appropriately. Two of the reranking strategies (SRR and IRR) need to have access to the RSVs of the retrieved SUs, but HRR only requires the handling of the two rankings.

3.3 Structural Query Expansion: CAS Queries

Another approach to personalization would be to perform a sort of query expansion but exploiting the structural characteristics of XML to build the expanded query. CAS queries allow us to make full use of the document structure, specifying in the query *what* we are looking for, and *where* this should be located in the required documents. The *what* involves the specification of the content, while the *where* is related to the structure of the documents. The general idea is therefore to transform the original CO query into an expanded CAS query somehow including the profile information. As far as we know, nobody else has ever used CAS queries in this way. In contrast to the previous approaches, this personalization strategy can only be applied to XML documents.

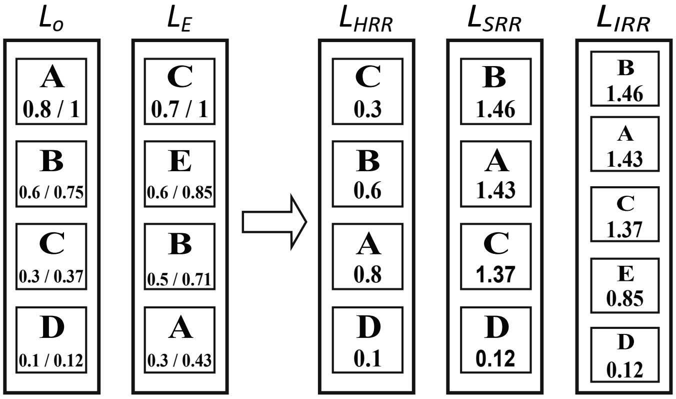

Fig. 1. Example of how the proposed reranking strategies work. The numbers associated with each SU correspond to its original/normalized RSV.

In order to allow CAS queries to be specified, we have selected the NEXI language [49], widely used within INEX. The general form of a NEXI CAS query is *//A[B]//C[D]*: "returns C descendants of A, where A fulfills the condition B and C fulfills the condition D". A and C are paths specifying structural restrictions, whereas B and D are filters specifying content restrictions, and *//* is the descendant operator. C is the target path (the last structural unit in C is the one we want to retrieve) and A is the context. Each content restriction will include one or several *about* clauses, connected by either *and* or *or* operators. Each *about* clause contains both a set of terms and a relative path from the structural unit which is the container of the clause, to the structural unit contained in it where these terms should be located. NEXI is both a simplified XPath containing only the descendant operator in a tag path and an extended XPath containing the *about* clause[8].

For example, the following CAS query attempts to retrieve chapters dealing with personalization and containing a bibliography of INEX, within books with a title related to information retrieval:

```
//book[about(.//title,information retrieval)]//
  chapter[about(.,personalization) and about(.
  //bibliography,INEX)]
```

In this case, the chapter units are the target and the

[8] The *about* clause is the IR counterpart of the classical *contains* clause used in XPath, which requires an exact matching between the textual content of the clause and a part of the text in the structural element being evaluated.

book units are the context.

We are going to transform the original CO query into a CAS query in such a way that its target part coincides with the original query and its context part contains the profile information. As the original query does not specify any structural restriction, we use in the target part the NEXI path wildcard operator "$*$" (meaning first or subsequent descendant), so that `//*[about(.,originalQueryTerms)]` is a CAS query equivalent to the original CO query. For the context part of the query, we propose to use the largest retrievable structural unit in the collection, `MaxUnit` (which is the less restrictive SU to hold the profile terms). Therefore, the expanded CAS query would be

```
//MaxUnit[about(.,profileTerms)]//*[about
  (.,originalQueryTerms)]
```

Instead of using all the profile terms together, another option is to let each term be part of a different *about* clause, all of these clauses being connected by the *or* operator. The motivation behind this modification is that, usually, a keyword query has an implicit conjunctive semantics, but in our case it is not necessary that all the profile terms have to appear in the context part of a relevant SU. This new version of the expanded CAS query is then

```
//MaxUnit[about(.,profileTerm1) or about
  (.,profileTerm2) or...or about(.,profile
  TermK)]//*[about(.,originalQueryTerms)]
```

3.4 Modification of the Retrieval Model

All the personalization strategies considered so far try, in some way, to separate the contributions of the original query terms and the user profile terms. They do it externally, out of the underlying retrieval model implemented by the search engine. Now, we are going to propose an internal modification of the retrieval model ranking method, which also points in the same direction. This is not very common practice in personalization strategies (specially in web personalization, where the search engine cannot be modified, mainly because it is usually unreachable to the researchers).

This strategy depends completely on the retrieval model underlying the search engine being considered. In this case, we have used Garnata, which is an IRS based on probabilistic graphical models, namely Bayesian networks and influence diagrams. A description of the theoretical basis of this system can be found in [5], [6]. Nevertheless, it is possible that the ideas underlying this modification of Garnata can be applied to other systems.

To understand how we have modified Garnata's search engine it is necessary to briefly explain how it computes the RSV of each SU in a document. It combines two different types of information. On the one hand, the specificity of the SU with respect to the query: the more terms in the SU which appear in the query, the more relevant the SU becomes. That is to say, the more clearly the SU is only about (at least a part of) the topic of the query. On the other hand, the exhaustivity of the SU with respect to the query: the more terms in the query which match with terms in the SU, the more relevant the SU is, i.e., the more clearly the SU comprises the topic of the query. The SUs which best satisfy the user information needs expressed by means of the query should be, simultaneously, as specific and exhaustive as possible.

These two dimensions of the relevance of an SU with respect to the query are calculated in a different way. To compute the specificity, the probability of relevance of each SU, given the query, is obtained through an inference process in the Bayesian network representing the structured document collection. The exhaustivity is obtained by first defining the utility of each SU as a non-linear transformation of the proportion of the terms in the query that appear in this SU. Then the Bayesian network is transformed into an influence diagram which computes the expected utility of each SU, by combining the probabilities of relevance and the utilities in a principled way.

Essentially, the utility of each SU U given a query Q is defined as

$$util_{Q,n}(U) = nidf_Q(U)\frac{e^{(nidf_Q(U))^n} - 1}{e - 1},$$

where $nidf_Q(U) = \frac{\sum_{t\in U\cap Q} idf(t)*w(t|Q)}{\sum_{t\in Q} idf(t)*w(t|Q)}$ is a kind of normalized inverted document frequency of the terms appearing in U and Q, which increases with the number of terms in $U \cap Q$; $w(t|Q)$ are the weights associated to the query terms; n is a parameter that controls (in a non-linear way) the extent to which more terms from the query must be contained in an SU in order to get a high utility value for this unit. In this way, the greater the value of the integer parameter n, the more similar the behaviour with respect to a strict AND operator.

When using expanded queries, which are composed of the terms appearing in the original query and the terms coming from the profile, the problem is that all of these terms are used to compute the utility $util_{Q,n}(U)$. Therefore, the terms from the profile still have a great influence (despite their lower weights), possibly distorting the original query. For example, considering a query composed of 4 original terms and 20 profile terms, Garnata would possibly prefer to return an SU having only 1 original term and 15 profile terms, instead of an SU with all 4 original terms and 5 profile terms.

To avoid this problem, although all terms (original and expanded) are still used in the computation of the specificity (probability), only original terms are used in the calculation of the exhaustivity (utility). This modification of the retrieval model can be used together with the normalized query expansion strategy, and also with reranking; within the CAS queries approach it makes no sense because the original and profile terms are not used together, they are used separately in the target and context subqueries of the complete CAS query.

4 Experimental Methodology and Results

This section shows all the necessary components to perform the evaluation process of the different personalization strategies. It also shows the obtained results and the main conclusions.

As we have already mentioned, we will use Garnata as our structured IRS. Garnata has been tested at three editions of the INEX Workshop ([10] describes our last participation). It has also been applied to build a real IRS for parliamentary documents [11].

4.1 Document Collection, Queries and Relevance Assessments

The INEX initiative has provided the XML IR community with a wide range of XML test collections for evaluating different models and approaches in the tracks offered in each campaign. However, in the case of evaluating XML personalization strategies, there is a total lack of such collections (this situation also applies to plain documents). Therefore, in order to evaluate their proposals, researchers are obliged to create their own test collection.

For this reason, in our experiments we have used a document collection composed of a subset of the Records of Parliamentary Proceedings of the regional Parliament of Andalusia in Spain (marked up in XML), corresponding to different Committee Sessions. Each Committee is devoted to a specific area of interest, e.g. agriculture or education. The document collection is composed by 658 committee session documents. Each of these documents is the transcription of the speeches of the members of the Parliament who talk in the given committee session. All documents belong to the sixth and seventh terms of office (containing a total of 432,575 different SUs) and having a size of 122MB.

We have used a set of 23 different queries (some examples are given in Table 1). These queries have been formulated by real users of the cited document collection. Hence, they represent a small but trustworthy sample of real user information needs.

TABLE 1
Some examples of the 23 used queries (translated).

seville olive cultivation	musical activity
andalusian exports	economic expenditure scholarships
public employment	water purification
disease virus transmission	granada province investments

As we need relevance assessments in order to evaluate the performance of the different personalization strategies, we have carried out a user study involving 31 users. Each user was instructed to submit from two to four of the previous 23 different queries to the IRS. To evaluate the results, the user should assume his/her chosen profile corresponds to a person interested in documents related to the topics discussed in a specific Committee. Eight profiles were selected, corresponding to the following eight different Committees: Agriculture, Culture, Economy, Education, Employment, Environment, Health and Justice. Several users submitted and evaluated their queries by assuming (in turn) different profiles.

When a user evaluates a query under a given profile, a set of relevance assessments is obtained for this user, profile and query. We call the previous file of relevance assessments, an *evaluation triplet*. A total of $nt = 126$ different evaluation triplets were obtained. The reported results are the averages across these 126 evaluation triplets.

As it is time consuming for a user to evaluate a great number of the results returned by the IRS, the users were instructed to evaluate only the 50 first results returned by the system in response to a query. The problem is that, in this case, many possibly relevant results (for the given query and profile) would not appear among the first 50 results, so that these results would be considered as irrelevant. To alleviate this problem, the users evaluated the first 50 results returned by the system not only for the original query but also for the first 50 results returned by the Hard reranking personalization strategy. These evaluations were performed separately, at first place with the original IRS and next with the personalized IRS. After that, we have two lists of relevant results. In order to have a unique list for each evaluation triplet, we fuse these two lists, deleting duplicates and overlaps (maintaining the greater SUs in the latter case). It is important to note both that the user did not know which system was being used each time (in order to not being biased), and that there was no interaction between users. The user does not judge if a given retrieved SU is the best possible SU, but only whether or not its content is relevant (binary assessments) to the given query and profile.

4.2 Learning the User Profiles

As we have already mentioned, we use a weighted keyword-based user profile representation. User profiles have been learned from the corpus content. This is possible because the corpus is classified into different areas of interest, having learned a profile for the eight selected areas-committees. This article does not aim to determine the best profile, since it is more focused on the behaviour and retrieval performance of the different personalization techniques. Therefore, a simple approximation to build the user profiles has been initially considered.

The profile associated to an area of interest is comprised by those terms in the first k positions of the list of terms appearing in documents of this area, ordered by decreasing *tf*idf* and weighted by *idf*. *Idf* has been selected as the weight because each term is better represented by this value than by the *tf*idf* value, considering the full corpus. Fig. 2 shows an example of the user profile terms and weights learned from the Agriculture's Committee.

```
1.822*agriculture 1.535*sector 2.066*agrarian
2.068*fishing 1.965*production 1.659*aid
2.220*farmer 1.839*product 2.351*oil 2.098*rural
```

Fig. 2. Example of the first ten terms and idf weights corresponding to the user profile learned from the Agriculture Committee. The terms are translated to English and unstemmed.

We have also considered a different way of building the profiles. We have asked some expert users of the document collection to manually build the profiles. First, we provided them with the list of terms appearing in documents of each area of interest. Then the experts selected and ordered the terms that in their opinion better represented each area (although the weights are still based on idf).

4.3 Evaluation Method

In order to setup an evaluation criterion, we must specify that our objective is to evaluate the benefits of including the user profiles in the retrieval process. That is, to study the differences in performance obtained by using the proposed personalization strategies with respect to using the original query, in both cases considering the proper relevance assessments made by the users. This is different from the classical evaluation objective in XML retrieval, which is to identify the best possible SUs to return to the user. The idea is to measure whether the proposed personalization strategies will help the user to find the previously judged relevant components more easily, by comparing the results obtained with and without them.

For this purpose we consider particularly valuable the use of rank-based measures. We have chosen NDCG (Normalized Discounted Cumulative Gain) [25] as the evaluation metric, which better fits our requirements. This evaluation metric is designed for estimating the cumulative relevance gain obtained by a user examining the first documents in a retrieved list of results. Since users tend to check only the first results, a discounting factor is used to reduce the document effect over the metric value as its position increases within the ranking. The metric value for a given list of results, is calculated as follows:

$$NDCG@x = \frac{1}{N}\sum_{i=1}^{x}\frac{2^{rel(d_i)}-1}{\log(i+1)}, \tag{1}$$

where x is the evaluation threshold (50 in our experiments); i is the ranking position of the SU being evaluated; d_i is the SU at position i; $rel(d_i)$ is the relevance value of d_i; the normalization factor N is the DCG for the ideal ranking, where all the relevant results are consecutive, starting from the first position. With this normalization, the metric values are always between 0 and 1, making it possible to calculate averages among different evaluation triplets.

For plain documents the value of $rel(d_i)$ would be 0 if the document has been judged as irrelevant and 1 if it has been judged as relevant. However, considering that we are working with XML documents and the IRS can retrieve SUs of different granularity, two considerations must be taken into account in order to get the fairest evaluation results:

Overlap degree: if there is no overlap between the retrieved SU d_i and any of the SUs judged as relevant for the given evaluation triplet in the user study (the relevance assessments), then d_i will be considered as non-relevant, that is, $rel(d_i) = 0$. However, what does one do when an SU d_i overlaps with some of the SUs judged as relevant by the user (a match)? As the user did not judge all the possible SUs but only those which were retrieved by the system, it seems to us reasonable to assume that an SU which matches any relevant SU from the corresponding evaluation triplet is also relevant (to some degree). A rough approximation would be to assign $rel(d_i) = 1$ to any match, although we prefer to use a more refined approach. Possible options are, on the one hand, to calculate $rel(d_i)$ considering the overlap degree (in terms of text length) of the two SUs and, on the other hand, to use a function measuring the relevance in terms of the distance between the two SUs within the XML hierarchy.

In this study we follow the second approach. Let us explain the reasons for this choice. In our document collection there are four retrievable SUs: *proceedings* (the complete document corresponding to one session of a Committee), *initiative* (the debate of a parliamentary motion within a session), *intervention* (of a member of the parliament in the debate of a motion) and *paragraph* (of an intervention). Let us suppose, for example, that an initiative has been judged relevant by the user. The relevance degree of the proceedings where this initiative has been discussed should not depend on the length of the initiative, neither on the length of the rest of the initiatives in this session. In other words, all the initiatives are considered equally important to determine the relevance value of the proceedings. The same reasoning can be used with the rest of SUs. Therefore, as the exact size of the overlap between SUs is not important in this case, we consider the distance. Within the current XML hierarchy, the distance between two SUs can be 0 (exact match), 1, 2 or 3 (when an SU is a proceedings and the other is a paragraph). The value of $rel(d_i)$ is obtained as a function of the distance, as specified in Table 2.

TABLE 2
Values of $rel(d_i)$ as a function of the distance between SUs.

distance	0	1	2	3
$rel(d_i)$	1.0	0.7	0.4	0.1

NDCG structural normalization: after the application of any retrieval strategy to a given evaluation triplet, we will have a list of retrieved results and a list of their corresponding relevance assessments, which are used to compute the value of $NDCG@x$. The normalization factor, N, in equation (1), calculated as the ideal DCG value for the relevance assessments, is

$$N = \sum_{i=1}^{\min(x,rj)} \frac{1}{\log(i+1)}$$

where rj is the number of results judged as relevant by the user for this triplet. In this case $rel(d_i) = 1$ is always true because all the matches are exact and therefore the distance is always equal to 0. The only important quantity to determine the value of N is thus rj.

The problem with this normalization appears when either (1) an SU in the list of results matches more than one (say u) relevance assessment, or (2) several results (say v) match the same relevance assessment. In both cases the number of relevance assessments in the denominator and the number of SUs considered in the numerator of equation (1) is not coherent. In the first case, only one retrieved SU contributes to the summation of equation (1), but in contrast, u relevance assessments contribute to the calculation of N. This is not a fair situation, because although all relevant SUs have been retrieved (in a greater SU containing all of them, but they have been retrieved anyway), the contribution of these SUs to the calculation of the normalization factor N is penalizing the NDCG value. Retrieving an SU, greater than the ones which should be retrieved, is already penalized by the overlap degree and we should not penalize twice. To avoid this unfair situation we subtract $u - 1$ units from rj. In the second case the situation is similar but the other way around: v smaller SUs have been retrieved instead of a single greater SU. In this case we add $v - 1$ units to rj.

An example of the NDCG structural normalization calculation can be seen in Fig. 3. As we can see, there are $rj = 4$ relevance assessments in the example. The first result matches only one relevance assessment, so there is no problem in this case. The second result matches two different relevance assessments (so we subtract 1). Finally, the third, fourth and fifth results match the same relevance assessment (so we add 2). The final value of rj is then 5 (4-1+2).

Beside evaluating the retrieval performance of the different personalization techniques through the averages of the NDCG measures across all the evaluation triplets, it is also interesting to consider the robustness of these techniques. The ideal situation would be a method that never performs worse than using the original query, while often performing better using personalization. A simple measure of robustness, frequently used in pseudo-relevance feedback, is the Reliability of Improvement (RI) [39], also called robustness index [13]. In our context it is defined as the ratio of the difference between the number of evaluation triplets helped (n_+) and of those hurt (n_-) by the personalization method, to the total number of triplets, nt:

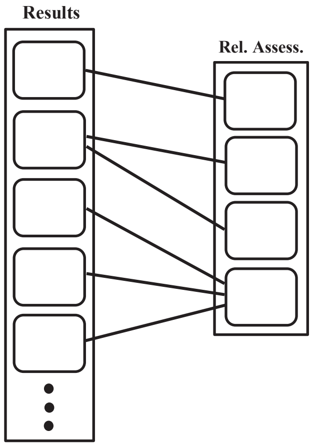

Fig. 3. NDCG structural normalization process.

$$RI = \frac{n_+ - n_-}{nt} \tag{2}$$

This measure varies from -1.0, when all triplets are hurt by the personalization method, to +1.0 when all triplets are helped.

4.4 Results

Tables 3 and 4 show the NDCG and RI values obtained from the different experiments when using the profiles generated automatically. The baseline result ($NDCG@50 = 0.400$) is obtained by using the IRS without any kind of personalization. The most basic personalization method (whose results could be considered another more advanced baseline), is to perform query expansion (QE) without using the weights of the profile terms (i.e. simply adding these terms to the original query).

The other personalization methods considered in Table 3 are normalized query expansion (NQE) and different reranking strategies: hard reranking (HRR), soft reranking (SRR), include reranking (IRR) and a modification of HRR where, instead of reranking the original query results using NQE (as HRR does), we rerank the results of NQE using the original query results. We call this modification inverse hard reranking (I-HRR). Its inclusion is motivated to test whether this strategy, which has been used in blind feedback, may be useful in personalization. We have also included another variation of HRR (called p-HRR), where we rerank the original query results using those obtained from a query composed uniquely of the profile terms. The idea is to illustrate the importance, in personalization, of reranking using the original+profile query instead of the profile query, by comparing the performance of HRR and p-HRR.

The methods considered in Table 4 are: the two versions of structural query expansion, one using all the profile terms within a single about clause (CAS) and the other using different about clauses connected by the *or* operator for each profile term (CAS-or); and the combination of the modification of the retrieval model with normalized query expansion and the reranking strategies (NQE+m, HRR+m, SRR+m and IRR+m).

TABLE 3
NDCG and RI values obtained in the experiments with QE, NQE, HRR, SRR, IRR, I-HRR and p-HRR, using the automatic profiles.

k	p_0	**QE**	**NQE**	**HRR**	**SRR**	**IRR**	**I-HRR**	**p-HRR**
NDCG@50								
5	0.33	**0.518***	**0.626***	**0.661***	0.603*	0.603*	0.408	0.331
5	0.66	**0.518***	0.575*	0.642*	0.626*	**0.623***	0.410	0.331
5	0.99	**0.518***	0.529*	0.623*	**0.631***	**0.623***	0.410	0.331
10	0.33	0.392	0.584*	0.644*	0.615*	0.613*	0.409	0.326
10	0.66	0.392	0.490	0.592*	0.612*	0.601*	0.412	0.326
10	0.99	0.392	0.419	0.540*	0.581*	0.564*	0.413	0.322
20	0.33	0.315	0.524*	0.607*	0.598*	0.591*	0.412	0.335
20	0.66	0.315	0.408	0.518*	0.574*	0.557*	0.414	0.335
20	0.99	0.315	0.345	0.473	0.543*	0.520*	0.418*	0.335
40	0.33	0.248*	0.419	0.527*	0.568*	0.553*	0.411	**0.341**
40	0.66	0.248*	0.317	0.449	0.524*	0.491*	0.416	**0.341**
40	0.99	0.248*	0.278*	0.417	0.496*	0.455	**0.425***	0.340
μ		0.368	0.459	0.558	0.581	0.566	0.413	0.333
σ		0.105	0.112	0.082	0.042	0.054	0.005	0.006
Baseline		0.400						
RI								
5	0.33	**0.246**	**0.532**	**0.579**	**0.635**	**0.635**	0.183	-0.159
5	0.66	**0.246**	0.365	0.444	0.595	0.579	0.214	-0.159
5	0.99	**0.246**	0.254	0.421	0.579	0.548	0.206	-0.159
10	0.33	0.048	0.444	0.484	0.587	0.571	0.183	**-0.127**
10	0.66	0.048	0.270	0.397	0.476	0.429	0.230	**-0.127**
10	0.99	0.048	0.103	0.302	0.421	0.373	0.254	-0.143
20	0.33	-0.151	0.294	0.373	0.468	0.437	0.246	-0.198
20	0.66	-0.151	0.040	0.190	0.389	0.341	0.262	-0.198
20	0.99	-0.151	-0.119	0.119	0.397	0.325	0.310	-0.198
40	0.33	-0.310	0.032	0.262	0.389	0.357	0.286	-0.183
40	0.66	-0.310	-0.198	0.024	0.294	0.230	0.310	-0.183
40	0.99	-0.310	-0.246	-0.016	0.183	0.040	**0.381**	-0.183
μ		-0.042	0.147	0.298	0.451	0.405	0.255	-0.168
σ		0.218	0.253	0.187	0.135	0.168	0.059	0.026

TABLE 4
NDCG and RI values obtained in the experiments with CAS, CAS-or, NQE+m, HRR+m, SRR+m and IRR+m, using the automatic profiles.

k	p_0	**CAS**	**CAS-or**	**NQE+m**	**HRR+m**	**SRR+m**	**IRR+m**
NDCG@50							
5	0.33	0.668*	0.675*	0.549*	0.561*	0.493*	0.493*
5	0.66	0.682*	0.686*	0.619*	0.643*	0.554*	0.554*
5	0.99	**0.687***	0.684*	0.645*	0.680*	0.583*	0.583*
10	0.33	0.659*	0.688*	0.583*	0.597*	0.523*	0.523*
10	0.66	0.681*	0.698*	0.650*	0.678*	0.578*	0.578*
10	0.99	0.685*	**0.702***	0.660*	0.701*	0.601*	0.601*
20	0.33	0.658*	0.681*	0.611*	0.628*	0.536*	0.536*
20	0.66	0.670*	0.691*	0.659*	0.692*	0.589*	0.589*
20	0.99	0.671*	0.692*	0.655*	0.705*	0.617*	0.617*
40	0.33	0.654*	0.673*	0.650*	0.668*	0.556*	0.556*
40	0.66	0.674*	0.676*	**0.682***	0.737*	0.606*	0.606*
40	0.99	0.678*	0.678*	0.680*	**0.738***	**0.628***	**0.628***
μ		0.672	0.685	0.637	0.669	0.572	0.572
σ		0.011	0.009	0.039	0.053	0.040	0.040
Baseline		0.400					
RI							
5	0.33	0.595	0.548	0.643	0.627	0.667	0.667
5	0.66	0.579	0.548	0.611	0.563	0.643	0.643
5	0.99	0.579	0.516	0.603	0.587	0.627	0.627
10	0.33	0.619	0.603	0.643	0.579	0.651	0.651
10	0.66	0.603	0.587	0.619	0.627	0.651	0.651
10	0.99	0.587	0.587	0.540	0.571	0.635	0.635
20	0.33	**0.635**	0.579	0.627	0.595	0.667	0.667
20	0.66	0.619	0.556	0.548	0.587	0.651	0.651
20	0.99	0.603	0.556	0.548	0.563	0.667	0.667
40	0.33	0.611	0.619	**0.651**	**0.675**	0.643	0.643
40	0.66	0.603	0.619	0.619	0.587	0.667	0.667
40	0.99	0.619	**0.635**	0.556	0.587	**0.714**	**0.714**
μ		0.604	0.579	0.601	0.596	0.657	0.657
σ		0.017	0.036	0.041	0.032	0.022	0.022

In all the cases, we have experimented with four different values (three for the case of expert profiles) of k, the size of the set of expansion terms (5, 10, 20 and 40). We have also used three different values for the parameter p_0 representing the normalization factor (0.33, 0.66 and 0.99) with all the personalization methods (except with QE, which does not use it). The tables show the NDCG and RI values obtained for the personalization methods for each of the 12 parameter combinations, as well as their average (μ) and standard deviation (σ). The best NDCG and RI values for each personalization technique appear in bold. For the NDCG values, if *both* a paired t-test and a paired Wilcoxon test detect statistically significant differences at level 0.01 with the baseline, we denote this by using "$*$".

Several conclusions can be drawn from our experiments. QE depends heavily on the number of terms in the profile. It only obtains better NDCG results than the baseline with very few terms, and deteriorates progressively as the number of terms increases. Even there are more evaluation triplets where QE loses than those where it wins, as the RI values show. NQE always gets better results than QE, although the tendency is the same: it is better to use few terms and a low normalization factor, in order to diminish the importance of the profile terms with respect to the original query terms. The reranking strategies HRR, SRR and IRR improve the results of NQE systematically (except for SRR and IRR using 5 terms and $p_0 = 0.33$) and always behave better than the baseline. Among these three strategies, the best NDCG result is obtained by HRR, although SRR and IRR are better on average. Moreover, it seems that SRR and IRR are somewhat less sensitive than HRR to an increase in the number of profile terms and normalization factor (as the lower standard deviations show). SRR and IRR also obtain much better values of RI than HRR, so that their behavior is more robust across different queries. However, the I-HRR strategy does not work: it is only slightly better than the baseline, although it is also quite stable with respect to the parameters k and p_0. The case of p-HRR is similar, although it always performs worse than the baseline.

The use of CAS queries produces very good results, which are always better than the corresponding results of all the previous strategies. Moreover, these results are much more homogeneous with respect to the number of terms in the profile and the normalization factor (exhibiting considerably lower standard deviations). This is an important advantage, because it guarantees good results independently on the number of profile terms (which may vary greatly depending on the specific situation). So, structural query expansion seems to properly manage the query-drift problem. From the two versions being studied, CAS-or is almost always better than CAS in terms of NDCG, although the opposite is true for the RI values.

Finally, the relation between the number of profile terms and normalization factor with the obtained performance, observed in NQE and the reranking methods, is completely reversed when we combine these strategies with the modification of the retrieval model: the performance is better as we increase k and p_0, and the NDCG and RI values are also more homogeneous. This change of tendency is positive, as we think that is more likely to find profiles composed of a high number of terms. Specially in the cases of NQE+m and HRR+m the NDCG and RI values are considerably better than their counterparts using NQE and HRR. In fact, HRR+m gets the best individual NDCG result (0.738) and the third best average (after CAS-or and CAS).

Tables 5 and 6 show the results obtained by using the profiles constructed by experts. It can be seen that the tendencies are practically identical when using automatic and expert profiles. In general, the results in Table 5 are slightly worse than those in Table 3, and the opposite is true for Tables 6 and 4. Therefore, it seems that only the best performing methods make the most to the more carefully constructed expert profiles.

Our conclusions in relation to what personalization strategy to recommend are: (1) if the search engine can be manipulated, then it may be a good idea to modify it (in the same way we have done with Garnata) in order to test its combination with the hard reranking strategy (using a high value of p_0), provided that the user profiles contain a high number of terms (ten or more); (2) if the search engine can manage CAS queries, then structural query expansion (also using a high value of p_0) is the recommended strategy (specially CAS-or); (3) otherwise, we would recommend using hard reranking if the number of terms in the profile is low (ten or less), and soft reranking otherwise (in both cases with a low p_0 value).

TABLE 5
NDCG and RI values obtained in the experiments with QE, NQE, HRR, SRR, IRR, I-HRR and p-HRR, using the expert profiles.

k	p_0	**QE**	**NQE**	**HRR**	**SRR**	**IRR**	**I-HRR**	**p-HRR**
				NDCG@50				
5	0.33	**0.445**	**0.600***	**0.626***	0.570*	0.570*	0.407	0.309
5	0.66	**0.445**	0.527*	0.602*	0.590*	0.586*	0.408	0.309
5	0.99	**0.445**	0.457	0.559*	0.578*	0.565*	0.408	0.309
10	0.33	0.346	0.549*	0.619*	**0.595***	**0.590***	0.408	0.327
10	0.66	0.346	0.437	0.549*	0.585*	0.568*	0.409	0.327
10	0.99	0.346	0.372	0.504	0.561*	0.531*	0.410	0.327
20	0.33	0.287*	0.481	0.584*	0.584*	0.570*	0.410	**0.343**
20	0.66	0.287*	0.359	0.487	0.550*	0.518*	0.413	0.342
20	0.99	0.287*	0.301*	0.444	0.518*	0.480	**0.416***	0.342
μ		0.359	0.454	0.553	0.570	0.553	0.410	0.326
σ		0.069	0.097	0.063	0.024	0.036	0.003	0.015
Baseline					0.400			
				RI				
5	0.33	**0.063**	**0.532**	**0.508**	0.548	**0.548**	0.151	**-0.167**
5	0.66	**0.063**	0.302	0.405	0.500	0.468	0.167	**-0.167**
5	0.99	**0.063**	0.127	0.310	0.437	0.405	0.183	**-0.167**
10	0.33	-0.056	0.349	0.437	**0.556**	0.524	0.175	-0.183
10	0.66	-0.056	0.095	0.278	0.437	0.357	0.214	-0.183
10	0.99	-0.056	-0.024	0.167	0.333	0.238	0.230	-0.183
20	0.33	-0.214	0.175	0.294	0.437	0.389	0.190	**-0.167**
20	0.66	-0.214	-0.024	0.222	0.325	0.230	**0.286**	**-0.167**
20	0.99	-0.214	-0.135	0.087	0.286	0.143	0.270	**-0.167**
μ		-0.069	0.155	0.301	0.429	0.367	0.207	-0.172
σ		0.121	0.210	0.133	0.097	0.139	0.047	0.008

TABLE 6
NDCG and RI values obtained in the experiments with CAS, CAS-or, NQE+m, HRR+m, SRR+m and IRR+m, using the expert profiles.

k	p_0	**CAS**	**CAS-or**	**NQE+m**	**HRR+m**	**SRR+m**	**IRR+m**
				NDCG@50			
5	0.33	0.622*	0.664*	0.513*	0.525*	0.473*	0.473*
5	0.66	0.637*	0.676*	0.678*	0.594*	0.521*	0.521*
5	0.99	0.640*	0.678*	0.615*	0.638*	0.550*	0.550*
10	0.33	0.649*	0.695*	0.567*	0.583*	0.511*	0.511*
10	0.66	0.666*	**0.706***	0.650*	0.675*	0.569*	0.569*
10	0.99	0.675*	**0.706***	0.658*	0.701*	0.597*	0.597*
20	0.33	0.665*	0.692*	0.639*	0.648*	0.546*	0.546*
20	0.66	0.684*	0.699*	**0.685***	0.722*	0.605*	0.605*
20	0.99	**0.692***	0.700*	0.677*	**0.730***	**0.637***	**0.637***
μ		0.659	0.691	0.620	0.646	0.557	0.557
σ		0.023	0.015	0.057	0.069	0.051	0.051
Baseline				0.400			
				RI			
5	0.33	0.500	0.540	0.587	0.571	0.619	0.619
5	0.66	0.468	0.532	0.587	0.508	0.619	0.619
5	0.99	0.468	0.516	0.556	0.540	0.603	0.603
10	0.33	0.579	**0.635**	0.627	0.611	0.643	0.643
10	0.66	0.563	0.603	0.587	0.603	0.635	0.635
10	0.99	0.548	0.603	0.556	0.595	0.635	0.635
20	0.33	**0.627**	**0.635**	**0.722**	**0.659**	**0.714**	**0.714**
20	0.66	0.587	0.619	0.635	0.603	0.690	0.690
20	0.99	0.603	0.619	0.587	0.603	0.698	0.698
μ		0.549	0.589	0.605	0.588	0.651	0.651
σ		0.058	0.047	0.052	0.044	0.040	0.040

5 Conclusions and Future Work

In this paper we have taken a step toward personalization strategies for the retrieval of XML documents, which is a relatively unexplored area of research. We have adopted a simple representation of the user preferences by means of user profiles composed of weighted terms. Then, we have focused on the development of methods which exploit these profiles in order to offer to the users those parts of XML documents that better reflect their preferences.

Our proposals include personalization methods to be applied both before and after submitting a query to the search engine, as well as within the retrieval process itself. In this way, we have studied simple query expansion and also more sophisticated and novel structural query expansion methods (both used to modify the original query before sending it to the search engine). We have also proposed several reranking strategies that transform the list of obtained results, after using the search engine to process the original query. These methods make use of the list of results obtained by an auxiliary expanded query (which includes the profile terms), instead of requiring a more complex processing (which usually needs to externally access the content of the documents and compare them with the profile). Finally, we have also considered internal modifications of the search engine, to better account for the different contributions of the original query terms and the profile terms, which may be used in combination with the other methods.

We have experimentally tested our methods by means of a user study on a parliamentary document collection marked up in XML, aiming to measure the benefits of managing the user profiles with these methods. To compare the results obtained with and without personalization, we have used two standard measures, the Normalized Discounted Cumulative Gain and the Reliability of Improvement, adapting them to the XML context.

Our experiments show that all the proposed methods significantly improve the baseline results (not using personalization) to a greater or lesser extent. In particular, because of their excellent results and robustness (in relation to the selection of some parameters, e.g. the number of profile terms being used), we can stand out structural query expansion and the combination of hard reranking with the modification of the retrieval model. These methods reach an NDCG improvement of 75% (71% on average) and 84% (67% on average), respectively.

As future work, we would like to study how to select the configuration parameters (i.e. the number of terms from the profile to use and the normalization factor of their weights) depending on the characteristics of the query, in order to obtain better personalized results. We also want to study the way of using other information included in the profile (other than search terms), as for example descriptors of a thesaurus [30]. In this paper we have focused on the personalization of the content part of the XML retrieval. Another interesting question would be to also consider the structural part (for example how to personalize CAS queries or how to manage structural preferences).

Acknowledgment

This paper has been supported by the Spanish "Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía", the "Ministerio de Ciencia e Innovación" and the research programme "Consolider Ingenio 2010" under the projects P09-TIC-4526, TIN2011-28538-C02-02 and MIPRCV:CSD2007-00018, respectively.

References

[1] J. Allan et al. Challenges in information retrieval and language modeling. Report of a Workshop held at the Center for Intelligent Information Retrieval, University of Massachusetts Amherst, 1-17, 2002.

[2] S. Amer-Yahia, I. Fundulaki, P. Jain, and L. Lakshmanan. Personalizing XML text search in PIMENT. Proceedings of the 31st International Conference on Very large Data Bases, pp. 1310-1313, 2005.

[3] S. Amer-Yahia, I. Fundulaki, and L. Lakshmanan. Personalizing XML Search in PIMENTO. Proceedings of the 23rd IEEE International Conference on Data Engineering, pp. 906-915, 2007.

[4] N.J. Belkin. Some(what) grand challenges for information retrieval. Keynote Lecture presented at the 30th European Conference on Information Retrieval, 2008.

[5] L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. Using context information in structured document retrieval: An approach using influence diagrams. Information Processing and Management, 40(5), 829-847, 2004.

[6] L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. Improving the context-based influence diagram model for structured document retrieval: removing topological restrictions and adding new evaluation methods. Lecture Notes in Computer Science, 3408, 215-229, 2005.

[7] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, and C. Martín-Dancausa. Content-oriented relevance feedback in XML-IR using the Garnata information retrieval system. Lecture Notes in Artificial Intelligence, 5822, 617-628, 2009.

[8] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, and C. Martín-Dancausa. Managing structured queries in probabilistic XML retrieval systems. Information Processing and Management, 46(5), 514-532, 2010.

[9] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, and C. Martín-Dancausa. A content-based approach to relevance feedback in XML-IR for content and structure queries. Proceedings of the International Conference on Knowledge Discovery and Information Retrieval, pp. 418-427, 2010.

[10] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, C. Martín-Dancausa, and A.E. Romero. New utility models for the Garnata information retrieval system at INEX'08. Lecture Notes in Computer Science, 5631, 39-45, 2009.

[11] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, C. Martín-Dancausa, A. Tagua-Jiménez, and C. Tur-Vigil. An integrated system for managing the andalusian parliament's digital library. Program-Electronic Library and Information Systems, 43(2), 156-174, 2009.

[12] L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, and E. Vicente-López. XML search personalization strategies using query expansion, reranking and a search engine modification. Proceedings of the 28th ACM Symposium on Applied Computing (SAC), pp. 872-877, 2013.

[13] C. Carpineto, and G. Romano. A survey of automatic query expansion in information retrieval. ACM Computing Surveys, 44(1), Article 1, 1-50, 2012.

[14] H. Chang, D. Cohn, and A. McCallum. Learning to create customized authority lists. Proceedings of the 17th International Conference on Machine Learning, pp. 127-134, 2000.

[15] G. Chernishev. Personalization of XML text search via search histories Proceedings of the SYRCODIS 2008 Colloquium on Databases and Information Systems, 2006.

[16] P.A. Chirita, C.S. Firan, and W. Nejdl. Summarizing local context to personalize global web search. Proceedings of the 15th ACM International Conference on Information and Knowledge Management, pp. 287-296, 2006.

[17] P.A. Chirita, C.S. Firan, and W. Nejdl. Personalized query expansion for the web. Proceedings of the 30th Annual International ACM SIGIR Conference, pp. 7-14, 2007.

[18] B.W. Croft, S. Cronen-Townsend, and V. Lavrenko. Relevance feedback and personalization: a language modeling perspective. Proceedings of the Second DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries, 2001.

[19] Z. Dou, R. Song, and J.R. Wen. A large-scale evaluation and analysis of personalized search strategies. Proceedings of the 16th International Conference on World Wide Web, pp. 581-590, 2007.

[20] N. Furh, M. Lalmas, S. Malik, and G. Kazai. Advances in XML Information Retrieval and Evaluation. Proceedings of the 4th Workshop of the INitiative for the Evaluation of XML Retrieval, Lecture Notes in Computer Science, 3977, 2006.

[21] N. Furh, M. Lalmas, and A. Trotman. Focused Access to XML Documents. Proceedings of the 6th Workshop of the INitiative for the Evaluation of XML Retrieval, Lecture Notes in Computer Science, 4862, 2008.

[22] T. Haveliwala. Topic-sensitive PageRank: a context-sensitive ranking algorithm for Web search. IEEE Transactions on Knowledge and Data Engineering, 15(4), 784-796, 2003.

[23] L. Hlaoua, K. Pinel-Sauvagnat, and M. Boughanem. Relevance feedback revisited: dealing with content and structure in XML documents. International Journal on Digital Libraries, 11, 1-24, 2010.

[24] W. Hsu, M.L. Lee, and X. Wu. Path-augmented keyword search for XML documents. Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, pp. 526-530, 2004.

[25] K. Jarvelin, and J. Kekalainen. Cumulative gain-based evaluation of IR techniques. ACM Transactions on Information Systems, 20(4), 422-446, 2002.

[26] G. Jeh, and J. Widom. Scaling personalized web search. Proceedings of the 12th International Conference on World Wide Web, pp. 271-279, 2003.

[27] J. Kamps, J. Pehcevski, G. Kazai, M. Lalmas, and S. Robertson. INEX 2007 Evaluation Measures. Lecture Notes in Computer Science, 4862, 24-33, 2008.

[28] G. Kazai, and M. Lalmas. INEX 2005 Evaluation Measures. Lecture Notes in Computer Science, 3977, 16-29, 2006.

[29] M. Lalmas. XML Retrieval. Morgan & Claypool Publishers, 2009.

[30] F. Liu, C. Yu, and W. Meng. Personalized Web search for improving retrieval effectiveness. IEEE Transactions on Knowledge and Data Engineering, 16(1), 28-40, 2004.

[31] C. Macdonald, and I. Ounis. Expertise drift and query expansion in expert search. Proceedings of the 16th ACM International Conference on Information and Knowledge Management, pp. 341-350, 2007.

[32] Y. Mass, and M. Mandelbrod. Relevance feedback for XML retrieval. Lecture Notes in Computer Science, 3493, 303-310, 2005.

[33] N. Matthijs, and F. Radlinski. Personalizing web search using long term browsing history. Proceedings of the 4th ACM International Conference on Web Search and Data Mining, pp. 25-34, 2011.

[34] L. Meister, O. Kurland, and I.G. Kalmanovich. Two are better than one! Re-ranking search results using an additional retrieved list. Technical report IE/IS-2009-01, Technion - Israel Institute of Technology, 2009.

[35] L. Meister, O. Kurland, and I.G. Kalmanovich. Re-ranking search results using an additional retrieved list. Information Retrieval, 14(4), 413-437, 2011.

[36] L. Nie, B. Davison, and X. Qi. Topical link analysis for web search. Proceedings of the 29th Annual International ACM SIGIR Conference, pp. 91-98, 2006.

[37] H. Pan. Relevance feedback in XML retrieval. Lecture Notes in Computer Science, 3268, 187-196, 2004.

[38] J. Pitkow, H. Schutze, T. Cass, R. Cooley, D. Turnbull, A. Edmons, E. Adar, and T. Breuel. Personalized search. Communications of the ACM, 45(9), 50-55, 2002.

[39] T. Sakai, T. Manabe, and M. Koyama. Flexible pseudo-relevance feedback via selective sampling. ACM Transactions on Asian Language Information Processing, 2(2), 111-135, 2005.

[40] R. Schenkel, and M. Theobald. Feedback-driven structural query expansion for ranked retrieval of XML data. Lecture Notes in Computer Science, 3896, 331-348, 2006.

[41] R. Schenkel, and M. Theobald. Structural feedback for keyword-based XML retrieval. Lecture Notes in Computer Science, 3936, 326-337, 2006.

[42] X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 824-831, 2005.

[43] A. Sieg, B. Mobasher, and R. Burke. Web search personalization with ontological user profiles. Proceedings of the 16th ACM International Conference on Information and Knowledge Management, pp. 525-534, 2007.

[44] B. Steichen, H. Ashman, and V. Wade. A comparative survey of personalised information retrieval and adaptive hypermedia techniques. Information Processing and Management, 48, 698-724, 2012.

[45] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. Proceedings of the 13th International Conference on World Wide Web, pp. 675-684, 2004.

[46] L. Tamine-Lechani, M. Boughanem, and M. Daoud. Evaluation of contextual information retrieval effectiveness: overview of issues and research. Knowledge and Information Systems, 24, 1-34, 2010.

[47] J. Teevan, S.T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. Proceedings of the 28th Annual International ACM SIGIR Conference, pp. 449-456, 2005.

[48] J. Teevan, S.T. Dumais, and E. Horvitz. Potential for personalization. ACM Transactions on Computer-Human Interaction, 17(1), article 4, 2010.

[49] A. Trotman, and B. Sigurbjörnsson. Narrowed Extended XPath I (NEXI). Lecture Notes in Computer Science, 3493, 16-40, 2005.

[50] X. Tao, Y. Li, and N. Zhong. A personalized ontology model for web information gathering. IEEE Transactions on Knowledge and Data Engineering, 23(4), 496-511, 2011.

[51] L. Zighelnic, and O. Kurland. Query-Drift Prevention for Robust Query Expansion. Proceedings of the 31th Annual International ACM SIGIR Conference, pp. 825-826, 2008.

Luis M. de Campos received his BSc degree in Mathematics in 1984. He completed his PhD. Thesis in 1988, researching on fuzzy measures and integrals and became Lecturer in Computer Science in 1991 at the University of Granada (Spain). He is currently a professor in the Department of Computer Science and Artificial Intelligence at the same institution. His current research interests include Bayesian Networks, Information Retrieval, Machine Learning and Numerical Representations of Uncertainty.

Juan M. Fernández-Luna obtained his Computer Science degree in 1994 from the University of Granada, Spain. In 2001, he got his PhD from the same institution, working on a thesis in which several retrieval models based on Bayesian networks for Information Retrieval were designed. Currently, his main research area is XML retrieval, working in collaboration with Juan F. Huete and Luis M. de Campos in XML personalization, collaborative IR, recommender systems and learning to rank. He has experience in organizing international conferences and workshops, e.g., I and II International Workshops on Teaching and Learning of Information Retrieval and SIGIR and CIKM workshops. He has been a co-editor of several journal special issues, He also belongs to the programme committee of the main IR conferences.

Juan F. Huete is an assistant professor at the Department of Computer Science and Artificial Intelligence at the University of Granada. He got his Ph.D. in 1995, researching on the uncertainty treatment in Artificial Intelligence under the formalism of Bayesian networks. From 1998, his research interest is Information Retrieval, designing retrieval models based on these graphical models. He is currently also working in the Recommender System field and other fields like collaborative IR or learning to rank. He has been a co-editor of special issues about Bayesian Networks and Information Retrieval, Teaching and Learning IR and Personalization.

Eduardo Vicente López received the BSc degree in Computer Science from the University of Almería in 2008, and the MSc degree in Computer Science from the University of Granada in 2011. He is currently working toward the PhD degree in the Department of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, Granada, Spain. His research interests include personalization, evaluation, structured information retrieval and user profiling.