UNIVERSITY OF GRANADA

Department of Computer Science and
Artificial Intelligence

PhD Program:
Information and Communications Technology

PhD Thesis:

PERSONALIZING XML INFORMATION RETRIEVAL

PhD Student:
Eduardo Vicente López

Advisors:
Prof. Dr. Luis M. de Campos Ibáñez and
Prof. Dr. Juan M. Fernández-Luna

Granada, December 2014

El doctorando D. Eduardo Vicente López y los directores de la tesis D. Luis M. de Campos Ibáñez y D. Juan M. Fernández Luna garantizamos, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección de los directores de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

The PhD student Sr. Eduardo Vicente López and the thesis advisors Sr. Luis M. de Campos Ibáñez and D. Juan M. Fernández Luna guarantee, by signing this doctoral thesis, the work has been done by the PhD student under the thesis advisors direction and as far as our knowledge reaches, performing the work, we have respected the rights of others to be cited, when their results or publications have been used.

Granada, Diciembre 2014.

Directores de la Tesis:

Fdo.: Luis M. de Campos Ibáñez Fdo.: D. Juan M. Fernández Luna

Doctorando:

Fdo.: Eduardo Vicente López

Agradecimientos

Hay muchas personas a las que quisiera agradecer su apoyo en la consecución de esta tesis doctoral.

En primer lugar, quiero agradecer a mis directores de tesis D. Luis M. de Campos y Juan M. Fernández-Luna. Sin lugar a dudas, este trabajo no se habría podido llevar a cabo sin su más que valiosa dirección, consejos y continua ayuda en todo aquello que ha sido necesario en cada momento. Os agradezco enormemente tanto el haberme dado la oportunidad de realizar esta tesis, como todo el tiempo dedicado en estos largos años.

En segundo lugar, me gustaría también agradecer a D. Juan F. Huete sus muy valiosas aportaciones en el trabajo y las publicaciones que avalan esta tesis. Igualmente, agradecer a todos los miembros del grupo de investigación *Uncertainty Treatment in Artificial Intelligence* la buena acogida recibida, desde el inicio de mi andadura en la universidad de Granada.

También me gustaría mencionar a todos mis compañeros de despacho, desde los inicios allá en el edificio Orquídeas hasta los más recientes del CITIC-UGR. Como sois muchos no os menciono individualmente, pero vosotros ya sabéis quienes sois. Os agradezco mucho esas charlas, el apoyo, las dudas resueltas, los buenos desayunos, e incluso los recientes partidos de pádel. Todo un placer el haber compartido todas estas experiencias con vosotros. Y también, a todos mis amigos 'de toda la vida', porque aunque ya no nos veamos tanto como nos gustaría, siempre estáis ahí.

Por otro lado, también me gustaría agradecer al grupo *Information Access*, del centro de investigación nacional CWI (Amsterdam), en el que realicé mi estancia internacional de doctorado. En especial a D. Arjen P. de Vries, como mi tutor de estancia, pero también a todos los miembros del grupo en general. Gracias por haberme acogido como uno más, por los consejos y las amenas charlas. Sin duda, no olvidaré esos *chocolate-breaks* y las reuniones semanales de seguimiento de todo el grupo.

Sin lugar a dudas, no quiero terminar sin hacer mención especial a mi familia, a mis padres y hermana. Gracias por haber sentado las robustas bases de lo que hoy soy, por vuestro incondicional cariño y constante apoyo. No habría llegado hasta

aquí, ni haber escrito estas líneas, si no fuera por vosotros. Y por último, a Alba, por quererme, por ser tan especial y por ser mi infatigable compañera de viaje.

Esta tesis doctoral ha sido financiada por la Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía y el Ministerio de Ciencia e Innovación, bajo los proyectos P09-TIC-4526 y TIN2011-28538-CO2-02, respectivamente.

Contents

I Introduction **1**

1 Introduction **3**

II Foundations **9**

2 Personalized Information Retrieval **11**

2.1 Introduction 11

2.2 User profiles 14

2.2.1 Information gathering 15

2.2.2 User profile representation 18

2.2.3 User profile update process 21

2.3 Personalization techniques 21

2.3.1 Before the search 22

2.3.2 Within the search 23

2.3.3 After the search 25

2.4 Evaluation 26

2.4.1 Evaluation of system-centred IRSs 27

2.4.2 Evaluation of user-centred IRSs 29

3 Structured Information Retrieval **35**

3.1 Introduction 35

3.2 Documents representation and the indexing process 37

3.3 Queries 43

3.4 Information retrieval models 46

3.5 Presenting results 50

3.6 Information retrieval evaluation . . . 51
3.7 The GARNATA IRS for XML retrieval . . . 55

III Research Contributions 59

4 Personalization Techniques for XML IR 61
4.1 Introduction . . . 61
4.2 Related work . . . 62
4.3 Developed personalization techniques . . . 65
4.3.1 Normalized query expansion (NQE) . . . 66
4.3.2 Reranking . . . 67
4.3.3 Structural query expansion: CAS queries . . . 69
4.3.4 Modification of the retrieval model . . . 71
4.4 Common experimental components . . . 73
4.4.1 XML document collection . . . 73
4.4.2 User study . . . 74
4.5 Experimental evaluation and results . . . 79
4.5.1 Evaluation metrics and structural adaptations . . . 80
4.5.2 Results . . . 83
4.6 Conclusions and future work . . . 94

5 An Automatic Evaluation Framework for Personalized IRSs 97
5.1 Introduction . . . 97
5.2 ASPIRE . . . 100
5.3 Related work . . . 104
5.4 ASPIRE use and validation . . . 107
5.4.1 Experimental framework . . . 108
5.4.2 Validation methodology . . . 110
5.5 Results . . . 114
5.5.1 User study-ASPIRE results comparison . . . 114
5.5.2 User study-ASPIRE Sieg et al. approach results comparison . 125
5.6 Conclusions and future work . . . 130

6 User Profiles **133**
6.1 Introduction . . . 133
6.2 Developed user profiles . . . 135
6.2.1 Building process . . . 137
6.3 Experimental evaluation and results . . . 141
6.3.1 Profiles based on terms . . . 142
6.3.2 Profiles based on subjects . . . 144
6.3.3 Profiles based on subjects and terms . . . 147
6.3.4 All user profiles results and conclusions . . . 148
6.4 Conclusions and future work . . . 156

IV Conclusions **159**

7 Conclusions and Future work **161**
7.1 Conclusions . . . 161
7.2 Future work . . . 164
7.3 List of publications . . . 165

References **168**

List of Figures

2.1 User profile build process main stages. 15
2.2 Main personalization techniques and places within the IR process where they are performed. 22
2.3 Classification of the main IR evaluation approaches. 27

3.1 A basic traditional IRS workflow. 37
3.2 An inverted file example. 40
3.3 A fragment of a sample XML document. 41
3.4 The XML document from Figure 3.3 as a tree. 42
3.5 An XML document as a tree. 45

4.1 Example of how the proposed reranking strategies work. The numbers associated with each SU correspond to its *original/normalized* RSV values. 70
4.2 DTD specification for the AP records of parliamentary proceedings, where committee sessions belong. 75
4.3 Example of NDCG structural normalization process. 83
4.4 NQE(+m) and HRR(+m) NDCG values from Tables 4.5 and 4.6. In the legends, the *yes* and *no* indicates whether the modification of the retrieval model has been used or not, respectively. 88
4.5 Same as Figure 4.4 but for SRR(+m)/≈IRR(+m) and *CAS*/*CAS-or* NDCG values. 89

5.1 Number of user study evaluation triplets relevance assessments (x axis) against ASPIRE evaluation triplets relevance assessments (y axis). Each point in the graph represents an evaluation triplet. 115

5.2 Histogram for Table 5.2 XML-based correlation values approach. . . . 118
5.3 NDCG user study-ASPIRE correlations (y axis) against the user study averaged NDCG values (x axis) for the Garnata (XML), BM25 and VSM results, respectively. 119
5.4 Averaged NDCG values from ASPIRE (y axis) and the user study (x axis), for each personalization technique-user profile configuration for the Garnata (XML) results. 120
5.5 Averaged NDCG values from ASPIRE (y axis) and the user study (x axis), for each personalization technique-user profile configuration for the BM25 and VSM results. 121
5.6 NDCG user study-ASPIRE_S correlations (y axis) against the averaged NDCG values from the real study (x axis), with $topkRel = 1500$. 127
5.7 Averaged NDCG values from ASPIRE_S (y axis) against the averaged NDCG values from the real study (x axis), for each one of the 149 personalization techniques-user profile configuration parameters combinations, with $topkRel = 1500$. 128

List of Tables

4.1 Two examples of the expanded final query using and not using *NQE*, where the original query terms are '*olive oil*', and NQE is applied with $k = 3$ and $p_0 = 0.66$ over the very low and very high user profile term weights, respectively. 67

4.2 The user study 23 queries (translated into English). 76

4.3 The first ten terms and *idf* weights corresponding to the eight selected user profiles. The terms are translated into English and unstemmed. . 77

4.4 Values of $rel(d_i)$ as a function of the distance between SUs. 81

4.5 NDCG and RI values obtained in the experiments with QE, NQE, HRR, SRR, IRR, I-HRR and p-HRR. 86

4.6 NDCG and RI values obtained in the experiments with CAS, CAS-or, NQE+m, HRR+m, SRR+m and IRR+m. 87

4.7 NDCG and RI values obtained in the experiments with QE, NQE, HRR, SRR, IRR, I-HRR and p-HRR using the expert profiles. 92

4.8 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. . 93

5.1 Averages and standard deviations of precision, recall and F metrics across the 126 evaluation triplets. 116

5.2 NDCG user study-ASPIRE Pearson correlation ranges, average values and standard deviations for the different runs. 117

5.3 Kendall τ correlations of the personalization techniques for each of the 12 combinations of the user profiles configuration parameters: number of expanded terms, k, and normalization factor, p_0. 123

5.4 Kendall τ correlations of the user profile configurations for each of the 13 personalization techniques. 124

5.5 Averages and standard deviations of precision, recall and F metrics across the 126 evaluation triplets, with $topkRel = 1500$. 126

5.6 ASPIRE_S: Kendall τ correlations of the personalization techniques for each one of the 12 combinations of the user profiles configuration parameters: number of expanded terms, k, and normalization factor, p_0. 129

5.7 ASPIRE_S: Kendall τ correlations of the user profiles for each one of the 13 personalization techniques. 130

6.1 Examples of the three proposed user profiles, for the 'agriculture and livestock' area of interest (unstemmed and translated into English). . 138

6.2 Final *tProf* and *sProf* user profiles using $exp[Terms|Subj] = 5$ and $p_0 = 0.66$. 139

6.3 Final *stProf* user profile using $expSubj = 2$, $expTerms = 3$ (to make it more clear and shorter), and $p_0 = 0.66$. 141

6.4 Maximum, average (μ) and std. (σ) performance values for the *tf*idf* and the *current* approach user profiles. 142

6.5 NDCG averaged values for the user profiles based on terms. 143

6.6 NDCG averaged values for the user profiles based on subjects. 144

6.7 NDCG averaged values for the user profiles based on subjects and terms (*stProf_add*). 149

6.8 NDCG averaged values for the user profiles based on subjects and terms (*stProf_max*). 150

6.9 NDCG averaged values for the user profiles based on subjects and terms (*stProf_addFill*). 151

6.10 NDCG averaged values for the user profiles based on subjects and terms (*stProf_maxFill*). 152

6.11 NDCG maximum, average (μ) and std. (σ) performance values for the six developed user profile approaches under the evaluation framework. Original (non-personalized) NDCG value: 0.388. '*' character shows the best user profile approach for each personalization technique, and '+' character shows the best personalization technique for a given user profile approach. 153
6.12 User profile parameters $k[-l]-p_0$ configuration for each maximum NDCG personalization technique-user profile performance, with '*' and '+' characters meaning the same as in Table 6.11. 153
6.13 General NDCG maximum (max), average (μ) and deviation (σ) values for each of the six proposed user profile approaches. 155

Part I

Introduction

Chapter 1

Introduction

Motivation

Not so long ago, people usually relied on other people, universal encyclopaedias, or went to libraries to get the information about the different things they wanted to know. The information was mainly stored in books, journals or any other kind of physical format. But increasingly since the advent of personal computers, and specially over the last few years with the proliferation of Internet, mobile devices, etc., and all their vast variety of new associated technologies, we are currently immersed in the digital era. Nowadays almost all the information is created and exchanged digitized, and its amount is increasing in an exponential way. It is obvious that a bigger amount of information is, in general, a good piece of new. Users will have more resources and places to search in order to satisfy their information needs, but this bigger amount of information is useless if users are not able to find the relevant documents which fulfil these information needs, due to this information overload.

Although the first steps in computerized information retrieval started in the late 1940s by Cleverdon [27], being the term *Information Retrieval* (IR) coined by Calvin Mooers on these days [83], was not until not so many years ago when the Information Retrieval Systems (IRS) became popular, especially the web based ones. The IRSs have actually become very important, since they represent the tool through most people search for any kind of information nowadays. Although most IRSs have been providing quite good results for the majority of users until now, if we join the previous huge rise of digital information, together with the fact that

users do not always specify accurately enough their information needs (they tend to formulate short and ambiguous queries), it is clear that the access to the relevant information is becoming more and more difficult every day.

Users generally search for information by submitting a query in natural language to the IRS. This IRS will provide the same output for a given query, independently of the user, since it only considers the query keywords as the representation of the user information needs. This issue is well-known as the *'one size fits all'* paradigm. Considering all the previous issues, a new approach is required, in which the user and not only the query, is considered as an important part within the retrieval process. *Personalization* [122] is this possible solution, being one of the key challenges and hot arising research areas within the information retrieval field [11, 128]. In this context, personalization may 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. Thus, personalization minimizes the information overload of users, making possible to better satisfy their specific information needs.

Another important consideration about the new generated information is that everyday is more common to store it around a well defined structure, which can be very useful in the retrieval process. The first steps in this relatively new research area, called Structured Information Retrieval (SIR), were given by Chiaramella in 2001 [23]. The main SIR asset is that it takes advantage of the documents internal structure, allowing to retrieve those specific parts of the document more related to the user information needs, e.g. a paragraph, instead of always returning the whole document, as traditional IRSs do. This feature is specially beneficial for users when dealing with big documents, e.g. books or this thesis itself, since they do not need to search the required information within them, but the structured IRS directly provides the more relevant parts. Or in other words, under these systems the documents are not considered anymore as atomic units of information to be retrieved as a whole. XML (eXtensible Markup Language) has emerged as the document standard for representing and exchanging this structured data.

This thesis is focused on personalizing this new and challenging SIR approach, as some problems appear when dealing with different parts of a document, e.g. the overlapping problem, but in most of the cases it is trivial to apply the developed

techniques to deal with flat documents, which would not happen the other way around.

Along this thesis we have used a document collection composed by documents in XML format from the Andalusian Parliament (AP) - the southern spanish autonomous region established in 1982. Our research group, Uncertainty Treatment in Artificial Intelligence (UTAI), has been cooperating with the AP since 2005. Along this collaboration they have been providing us with their two main official publications: the record of parliamentary proceedings and the official bulletins. It must be noted that the previous document collection has been used in most of the experiments carried out in this thesis, but of course, all the developed work is independent of this collection and it would perfectly works with any other appropriate document collection.

Main contributions of the thesis

All the contributions have been developed to work with structured information, such as XML documents. Work with structured documents is more difficult than with flat documents, but it also provides a series of benefits, e.g., a less effort for the user to find out the required information (the IRS retrieves specific relevant parts of a document instead of the full document), or much more powerful search capabilities for expert users (content and structure queries).

Any whole personalized process is composed by three main different stages: 1) how to gather and represent the information about the user in the user profile; 2) how to best use the previous user information within the retrieval process, in order to retrieve the closest results to the user interests and preferences, which best fulfil the user information needs; and 3) how to evaluate the performance of the whole personalization process. This thesis objectives are to develop different techniques for each of the previous personalized process steps. Next, we explain all the different developed contributions for all these stages.

Regarding to the user profiles, we have developed some different representations for them, based on the content of a set of documents the user has shown or is supposed to be interested in. For a better understanding it must be noted that the main structural unit (part) of these documents is the *initiative*, and that it has one

or more associated subjects manually assigned and extracted from the EUROVOC thesaurus[1]. There are three main different developed user profile approaches. The first approach is only based on the initiative subjects, being represented as a set of weighted concepts. The second approach is based on every term of the documents independently where they appear, in this case being represented as a set of weighted keywords. And finally, a third hybrid approach, where both subjects and terms are considered, being represented as a two level user profile, with a first level formed by subjects and a second level formed by terms. Our following publication [133] in the international PeGOV-UMAP conference may be checked, as the best published article for this part of the thesis contribution.

With respect to how to best use the information within the user profile in the retrieval process, we have developed a quite broad set of different personalization techniques. Any personalization technique is usually applied in one of the three different stages of the retrieval process: before, within, or after the search is actually performed. The best known approaches for each stage are a query modification, a modification of the retrieval model, and a reranking process, respectively. We have developed different personalization techniques covering all the previous three possibilities being even some of them hybridizations between the different retrieval stages. Our following publication [42] in the IEEE-TKDE journal may be checked, as the best published article for this part of the thesis contribution.

Finally, regarding to how to evaluate the performance of the personalization process, we have developed an Automatic Strategy for Personalized Information Retrieval systems Evaluation called ASPIRE. ASPIRE combines the advantages of system-centred approaches, together with the inclusion of the user context into the evaluation of the retrieval process, which is the main benefit of user-centred approaches. This is mainly possible based on the automatic relevance assessments generation, with the only prerequisite of having a pre-categorized document collection. ASPIRE avoids the difficulty and big associated costs of the interaction with real users, thus providing repeatable, comparable and generalizable results and conclusions. ASPIRE allows a completely automatic, fast and easy testing of any personalized IRS. Our following publication [134] in the UMUAI journal may be checked, as the best published article for this part of the thesis contribution.

[1] http://eurovoc.europa.eu/

Chapters overview

This thesis is organized in four different parts.

Part I includes an introductory Chapter 1 where the motivation, main contributions and this overview section of the thesis are presented.

Part II includes two different chapters with the foundations of both the Personalized and Structured Information Retrieval fields. These chapters explain the basics to better understand and see the *big picture* of the Research Contributions presented in Part III.

Chapter 2 gives an overview of Personalized IR (PIR), starting with a glimpse of traditional IR and followed by the different steps and classifications present on any personalization process, i.e., how to get the information about the user, how to represent it to build a user profile and how to update the profile. Then, it presents the different personalization techniques classified based on which part of the retrieval process they are performed. And finally, a section about the different approaches on how to evaluate the personalization process.

Chapter 3 gives an overview of the general concepts for traditional IR and their adaptations for structured IR. The chapter sections will present the different stages in the design of any IRS, such as, how to represent documents, build an index based on them, and how the queries represent the user information needs. How different IR models try to match queries and documents using the built index, different ways to present the retrieved results, and how to evaluate the whole process. Finally, the last section is devoted to briefly explain Garnata, our IRS for XML retrieval, which will be used for this thesis experiments.

Part III includes three different chapters with the Research Contributions of this thesis.

Chapter 4 explains with detail all developed personalization techniques including an exhaustive experimental evaluation and the obtained results. Within the experimental evaluation section, the used XML document collection and the carried out user study from where we got the relevance assessments later used for the evaluation, are also explained. These two components will also be useful for the following chapters.

Chapter 5 describes ASPIRE (an Automatic Strategy for Personalized Information Retrieval systems Evaluation). This strategy is able to evaluate any personalization technique with almost no effort and cost. It is an alternative to the costly user studies, joining the advantages of the system-centred and user-centred evaluation approaches. ASPIRE itself is evaluated under three different retrieval models, using the previous chapter personalization techniques, and justify one of its automatic relevance assessments generation criteria comparing itself against another state-of-the-art approach, which does not consider these criteria.

Chapter 6 shows a set of different ways to build and use user profiles, based on subjects, terms, and a hybrid approach among the two previous. These user profiles will be based on the content of the documents a given user is or is supposed to be interested in. We will show how well each of these user profiles performs and some derived extra benefits.

Part IV includes a final Chapter 7 describing the thesis general conclusions, the possible future work, and the list of publications which support this thesis.

Part II

Foundations

Chapter 2

Personalized Information Retrieval

2.1 Introduction

The term Information Retrieval may be used in many different contexts with different meanings. According to Manning et al. [76], within the information science field, it might be defined as:

> Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).

As may be expected, before the appearance of the duple computers-internet and its democratization, the amount of digital information was very limited. The use of IRSs was mainly focused on searching where the required information was stored in physical format. Thereby, IR was an activity only a few people used to be involved in. It was mainly used within professional environments such as scientific, law or medical catalogues or within libraries for the rest of people. In those days (approximately from late 1940s to late 1980s), the search was always based on author, title or keywords, being more a database style search rather than a real IR search style, with all its associated potential we currently know.

However, nowadays we are immersed in a world where the possession and the accurate management of information is crucial. With different implications, this affirmation is true for all levels of our society ranging from governments, companies

or organizations to every single citizen. We currently live in the so-called *'knowledge society'*.

Digital information allows easy and efficient storage, access or modification processes over it. Thus, almost every existent non-digital information source, as might be music, video, books, document collections, etc., are being progressively digitized, and of course, all new created information is digital. This fact is leading to an exponential increase of digital information in recent years, especially owing to Internet. Thus, the access to relevant information is more and more difficult everyday.

In order to be able to find relevant results within this huge amount of information, the use of IRSs has become almost a must, being very common and daily used by millions of people, specially the web based search engines. As the IRSs have become the most dominant way of information access for most people, the IR field has experimented a high attraction by both, the private and research communities.

Roughly speaking, there are two basic evaluation metrics to measure an IRS retrieval effectiveness: *precision* (proportion of retrieved relevant documents from the total number of retrieved documents) and *recall* (proportion of retrieved relevant documents from the total number of relevant documents in the collection). The higher both metrics are, the better is the IRS retrieval model, but both metrics tend to be inversely proportional, so a trade-off between them is needed.

Even though IRSs have become very useful to search for information, this kind of systems retrieve results based only on the information contained in the user issued query, which almost always is composed of a small set of keywords in natural language. This strategy has achieved good retrieval results in the last years. Nevertheless, the IRS returns the same results for the same query independently of the user, which is well-known as the *'one size fits all'* problem. To overcome the previous problem, also considering that user queries are usually short and ambiguous [120], together with the previously cited exponential increase of information, a new approach is required.

The first approach to avoid the previous problem was to use some additional information about the relevance of the results of an issued query. There are two main approaches: Relevance Feedback (RF) if the feedback (relevant results) is provided by the user, and pseudo-RF if the feedback is automatically inferred from the first

query retrieved results. The classical algorithm for the relevance feedback methodology is the Rocchio algorithm [76]. With the provided feedback, these techniques refine the original query to best capture the actual user intent, because users normally do not know how to accurately express an information need. The RF process may be repeated iteratively until the user is satisfied with the results. According to Croft et al. [30], these approaches may be considered as barely short-term *'personalization'* approximations.

RF usually gets good retrieval performance, mainly because the user itself specifies which results are relevant, and those are used to refine the original query. In general, RF techniques improve precision, but specially recall. Unfortunately, in real world environments users are unwilling to make the extra effort to provide these relevant results for feedback [66]. Pseudo-RF was designed as an alternative to RF to avoid this problem, but lower retrieval effectiveness is accomplished, since some of the first query results may not be relevant for the user.

The two previous RF approaches were the first attempt to adapt the results to the user, but the user itself need to be considered as an important part within the retrieval process. A new approach, in which the user personal information is stored and ready to be used on any submitted query, without place any burden on the user is still needed. *Personalization* [51, 45, 122] is this new approach where the user is very important in the retrieval process, and it has become a hot arising research area [11, 128]. In the IR field, personalization may be defined as:

> Personalization is the process by which, using information about the user (generally stored in a user profile) and the issued query, the most relevant results are provided with respect to the user interests and preferences, minimizing the information overload, and making possible to better and faster satisfy the user information needs.

However, there is a wide variety of environments where personalization may be applied. Early personalization research focused on providing filtering or rating systems for different applications such as email [75] or electronics newspapers [22]. Later, personalization was also applied on improving navigation effectiveness with the help of browsing assistants, such as Letizia [71] or WebMate [20], and adaptive web pages [97]. But, as IRSs have become so important in the search of relevant

information for most users almost everyday, the main application of personalization is precisely in these systems, especially within the web based IRSs [73, 55, 81], trying to improve the IRSs retrieval effectiveness. Even exploring the state-of-the-art references we have found a scientific publishing web [60], where the next message appears *"To use the personalized features of this site, please* `log in` *or* `register`*"*, being a perfect example of the personalization necessity and importance nowadays.

Focusing again in this thesis topic, i.e. IRS personalization, the introduction of personalization into the retrieval process allows some potential advantages, such as the disambiguation of short queries, the increase of top results precision, the improvement of domain specific retrieval tasks, and even, the inclusion of social behaviours, as it happens with recommender systems.

Any IR personalization process has three main different stages: 1) to acquire and represent the user interests and preferences in the user profile, 2) to exploit the best as possible the user profile information within the retrieval process, and 3) to evaluate the whole personalization process. We may consider some additional issues, such as privacy in the personal data collection and management process [68], or different ways to present the personalized results [2], with the intention of presenting this information to the user in the most easy and intuitive way.

The rest of the chapter is dedicated to explain, in a detailed way, the previous three main stages of any personalization process.

2.2 User profiles

We may briefly define a user profile as a data instance of a user model, which tries to best represent a given user.

Most IRSs are evolving from the system-centred IRS to the user-centred IRS approach. Since the former do not consider the user at all within the retrieval process, they provide the same output for everyone, given a query. However, the latter adapt their output to the user, considering the user profile information.

Personalized search may be considered as a subset of the broader field of contextual search. Tamine et al. [125] article shows a good classification of the different contextual dimensions under five main categories: device, spatio-temporal, user context (personal and social), task-problem and document context. In practice, it is

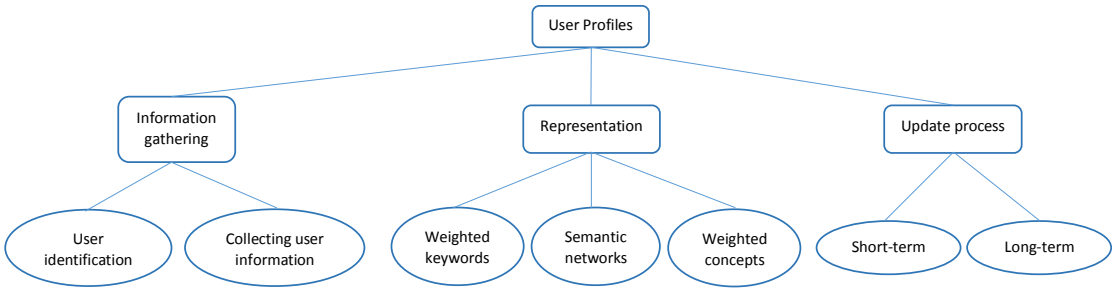

Figure 2.1: User profile build process main stages.

too much difficult to join all these contextual dimensions, and most research articles only focus on the user or task-problem context. However, with the current smart devices increase, and thanks to their many sensors, also the device and spatio-temporal contextual dimensions are being used.

In our case, we will focus on the context of users, defined by their interests and preferences. The main objective of any personalization process is how to best represent and exploit this user context. The intention is to fit the results to the users the best as possible, thus better and faster satisfying their information needs.

The quality of personalized results will highly depend on the user profile quality and how it is exploited in the retrieval process. Hence, the user profile build process is a very important step in order to obtain good personalized results, but at the same time very difficult, since user interests and preferences are difficult to be captured and they also change over time [70, 95].

Next, we are going to show the three most important stages in the user modelling-user profile build process (see Figure 2.1 for a graphic glance), according to Gauch et al. [48].

2.2.1 Information gathering

The first step to build a user profile is to get all the possible or needed information about the user. To do this, users must be uniquely identified by the system. This gathering process can be done either on the user device or on the server itself. Depending on where the process is performed the available information will be different. This information may be collected explicitly, introduced by the user, or implicitly, normally by a software agent. In general, implicit data collection places

no burden on the user, and as these systems perform the same as or even better than explicit systems, their use is preferred.

User identification A user identification is required to build an individual user profile. The three main approaches for user identification are cookies, logins, and software agents. Each approach has its advantages and disadvantages.

Cookies. Their main advantage is that their use is very easy and totally transparent for the user, being quite effective in some cases. But at the same time they have some disadvantages, such as: if the user accesses the IRS from different devices the cookies will be different, and therefore, the profiles. In addition, if the device is used by more than one person, there will be an unique inaccurate cookie-profile for all users. Furthermore, as the use of cookies is transparent for users, and they are able to allow or deny their use and even delete them, a given user may block the creation of the user profile or even delete it, likely without being aware of it.

Logins. Their main advantage is that a login process is directly done by the user, so there is no room for identification mistakes, providing user profiles with a much better accuracy and consistency than cookies. Another advantage is that the user will have the same profile independently of the used device. The main disadvantage is that users must be convinced to register and login each time they want to use the IRS.

Software agents. A software agent is a small program which users must install on their computers. These programs track every user interaction with the IRS, but not only this, they also may check their emails, bookmarks, or even track their eyes movements. Their main advantage is that they are the most accurate and reliable approach, since they have been programmed specifically for this purpose, being able to capture much more information than the other approaches. Their also important disadvantage is that users must install these programs on their devices, knowing that they are going to be highly tracked. Normally, users do not like to be tracked and feel spied, so this approach is almost no used.

Actually, the best compromise would be to use logins, but also providing the possibility to use cookies, for those who do not want to register and login each time they use the system.

Collecting user information. The user information collection may basically be performed by two different ways: explicit or implicitly provided by the user.

Explicit user information collection. To collect the information about the user explicitly, this user must want to voluntarily give this information, usually through forms and questionnaires. Any kind of information may be collected, such as, the user gender, birthday, city, marital status, interests, etc. But this information collection need to be done carefully, since if too much data is required, the user will be less prone to provide it. Hence, a trade-off between quantity and quality with the user required information must be taken, because most of the times, users are not so proactive, have not enough time, or are worried about their privacy to provide their personal information. If users feel swamped and do not provide any information, no user profile can be built for them. Therefore, this approximation is barely recommended nor lately very used.

The first approximations following this approach were some sites with customizable interfaces. They collect the user preferences in order to provide adaptable services to improve the information accessibility, as MyYahoo! [86] or iGoogle [58]. These webs organize and adapt their content based on the user preferences. Another example is [112], more focused on navigation. This system is a customizable and intelligent interface for webs, which assists the user to find relevant information.

As it is already stated before, user interests and preferences are difficult to be captured or expressed, also changing over time [95]. Considering the first part of this statement, another disadvantage for the systems which collect the user information explicitly is that, users are not always able or proactive to accurately provide their own information, conscious or unconsciously, probably because they also do not expect very good results from personalization systems yet. This fact is something this thesis and other works try to amend, because personalization is normally very useful and beneficial for users. Additionally, and considering the second part of the above statement, the already constructed user profiles remain static, unless the system frequently asks users to update their own information. This fact will definitely burden users, and their associated user profiles will be likely outdated and inaccurate sooner than later.

Implicit user information collection. There are several implicit approaches which try to avoid the previous explicit approach problems. Their main advantage is that they do not require user intervention, while they can be actively collecting information about the user all the time. Therefore, the user profile may be always accurate

and updated with no effort from the user. The main disadvantage of implicit user information collection is that it does not allow to collect robust enough negative feedback.

User browsing histories are a first implicit user information collection approach. They usually contain the user visited urls, date and duration of these visits. User interests may be inferred from this data, thus being able to provide personalized services. Its main disadvantage is that this browsing history is related to a single user computer. Two different examples of this approach are [10, 129], where users must configure a proxy server acting as their gateway to Internet, capturing this way all the user traffic.

Another implicit user information collection approach are the automatic agents, which are commonly used to track user interactions with the IRS, collecting their interests and preferences interactively while they browse. These agents are an independent installed application or a plug-in for a browser. This approach is able to collect a much more richer set of information than browsing histories. Additionally to the user browsing histories capabilities, they also can collect other sources of information such as the favourites and downloads made by the user. *Letizia* [71] was one of the first implicit approach systems which, based on the already user visited pages and bookmarks, recommends possible links of interest within the current page for the user. *Let's browse* [72] was an extension of Letizia for collaborative browsing suggestions. Another example is UCAIR [113], where implicit feedback information is used to perform query expansion based on previous queries, and instant result reranking based on clickthrough data.

In general, explicit approaches were used first, since they were more accurate and obtained better personalized results. But implicit systems have improved gradually and demonstrated to build profiles, at least as good or even better than explicit approaches, according to Teevan et al. [127].

2.2.2 User profile representation

In the previous section 2.2.1, we have seen how to gather the needed user information to build the user profile. Once we have this data, we need to define a way to represent it. This information representation will be stored in the so-called *user profile*. This

user profile will be used by any personalization technique in order to retrieve closer results to the user interests and preferences. According to Gauch et al. [48], there are three main representations for user profiles: weighted keywords, semantic networks, and weighted concepts.

Weighted keywords. This is the most common user profile representation, the simplest to build, and one of the first explored approaches. They require a big amount of user feedback in order to learn all the terms, by which any user interest is represented, and to be able to match this interest-terms with the future retrieved documents. The keywords and their associated weights may be automatically learned from the user visited documents or directly given by the user. The keyword weights show the importance of each keyword within the profile. The main problems of this kind of user profiles is the keywords synonymy (different words with the same or similar meanings), which may result in a recall decrease, and keywords polysemy (same word with different meanings), which may result in a precision decrease. These problems may make this kind of user profiles somewhat ambiguous. Examples of this user profile approach are Sugiyama et al. [123], where they build three different user profiles based on relevance feedback and implicit information, user browsing history, and a modified collaborative filtering. Other examples are Amalthaea [85], where Moukas learns the user profiles from the user visited web pages, based on the well-know *tf*idf* approach, and WebMate [20], where Chen et al. build user profiles formed by a vector of keywords for each user area of interest.

Semantic networks. Within this kind of user profile representation each node represents a concept. The semantic network helps to avoid the previous weighted keyword representation synonymy and polysemy problems, but they must learn the terminology (terms) associated to each concept. Examples of this approach are [49], an online digital library filtering system, where Gentili et al. initially have a semantic network of unlinked concept nodes. Each concept is represented with a single and representative term for that concept. As the user profile is enriched within the learning process, more weighted terms are associated and linked to the corresponding concepts, being the latter ones also linked between them. Another example is [82], where a filtering interface is created to personalize the results from the Altavista search engine. The user profiles are composed by three components: a header, including the user personal data, a set of stereotypes and each stereotype interests.

Another semantic network example is [115], where a personalized search system with ontology-based user profiles is presented. These user profiles are built assigning scores to user interests, implicitly derived from concepts of the ODP ontology[1]. Since the user interests are dynamic, a propagation algorithm is used to keep these interests updated. Finally, Tao et al. [126] propose a personalized ontology model for knowledge representation and reasoning over user profiles. This personalization model learns ontology-based user profiles from both a public knowledge base and a user local base information. Most of the previous studies have focused on one or another but not on both sources of information. This model is evaluated by a comparison study against a set of benchmark models in web information gathering.

Weighted concepts. They are similar to the semantic networks, since they also have conceptual nodes and relations between them, but in this case, the nodes are represented by abstract topics of interest for the user instead of terms. In contrast to the semantic networks, weighted concept user profiles are trained on examples for each concept a priori, already having the mapping between the vocabulary and concepts. This way these user profiles are robust to variations in terminology being learned with much less user feedback. At the same time, they are also similar to the weighted keyword user profiles, since they are usually represented as vectors of weighted concepts. Nonetheless, in the last few years it is common to use a hierarchical representation of concepts, usually derived from a taxonomy, thesaurus, or a reference ontology, instead of using concepts with no structure, allowing a much richer representation. An example of this approach is Trajkova et al. [129], where using concepts from the ODP ontology first three levels, they build user profiles based on the user browsing history. Another example is Vallet et al. [132], where the authors use weighted concept user profiles built over ontology-based semantic structures and metadata. They also build a context representation of the retrieval task, which is used to activate different parts of the user profile at runtime, thus matching the appropriate part of the user profile with the current retrieval task. Finally, Calegari et al. [15] show three different ways to use ODP: first, as a semantic support to find relations between concepts; second, identifying some ODP structure parts relevant to the user; and third, the user directly choose the ODP concepts

[1] http://www.dmoz.org/

he/she is interested in. After that, they study how to exploit these three user profiles, with personalization techniques based on query modification and re-ranking.

2.2.3 User profile update process

This step highly depends on the two previous steps, and it should be considered in their design. For example, the use of user information implicit acquisition techniques greatly facilitate this task. It is obvious that any user profile interests and preferences need to be updated recurrently, since they are dynamic and change over time [95]. A static profile is only useful in very specific cases. Therefore, an update process is needed in order to have an accurate and updated user profile.

Dynamic profiles could be separated into long-term user profiles (interests which define the user) and short-term user profiles (more related to a punctual information need). Therefore, both kind of profiles will evolve, but while the long-term will be rather stable, the short-term will be built and destroyed in a relatively short period of time. This short period of time is commonly related to a search session, which attempts to isolate a punctual information need or search intent.

Two examples where the authors try to combine both the long-term and short-term user profile approaches are Alipes [138], where three different vectors of weighted terms are used for each user interest, one for the long-term, other for the short-term (positive) and another for the short-term (negative). The other example is [33], where Daoud et al. study how to learn the long-term user interests aggregating the short-term user interests. The latter are related to some search activities delimited by a session boundary recognition system.

2.3 Personalization techniques

Once we have learned the user profile, we need to exploit its content the best as possible in order to provide the closest results to the given user interests and preferences. In this way, personalized IRSs will provide results with which the user will be more satisfied, because less time and effort will be required to cover the user information needs.

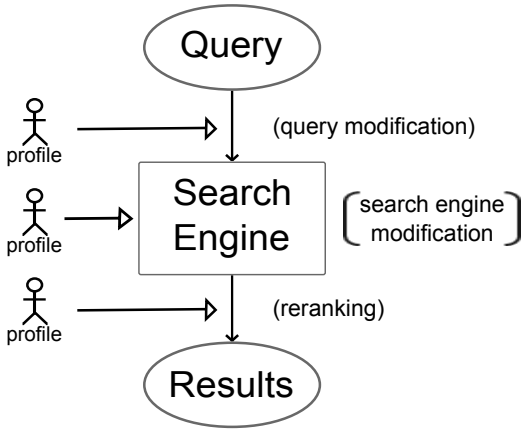

Figure 2.2: Main personalization techniques and places within the IR process where they are performed.

We are going to classify the different personalization techniques according to where they use the user profile information within the retrieval process: before, within or after the search is performed (see Figure 2.2).

2.3.1 Before the search

The most usual personalization technique before the search is a query reformulation. As the user query keywords are the only representation of the user information needs, these keywords should not be changed too much. Thus, the Query Expansion (QE) is one of the most natural, easy, common and successful techniques under this approach, which involves the expansion of the original query keywords with other additional terms.

Within personalization, this expanded query terms are the appropriate terms from the user profile. The user information need is represented by the original unexpanded query terms, and the expanded terms will represent the user interests and preferences. Therefore, the final expanded query will hopefully retrieve results which try to solve the user information need with results closer to the user, thus obtaining a higher user satisfaction degree with the system.

In general, when using query expansion recall is improved but sometimes at the expense of precision. Although, if we use a combined recall/precision measure, query expansion results in better retrieval effectiveness according to recent experimental

studies by Carpineto et al. [16]. Query expansion suffers the well-known *query-drift* problem [74, 142], which 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 for personalization, where the number of terms in the profile may be high and these terms can be highly unrelated to the original query terms. The most usual way of dealing with this problem, especially seen in feedback applications, is to emphasize the original query terms with respect to the expansion terms, for example giving less weight values to the expansion terms. Another approach is proposed by Parapar et al. in [94], where the authors try to optimize the non-relevant documents in the pseudo-relevance set to avoid the use of some harmful expansion terms, thus increasing the pseudo-relevance feedback robustness.

Some works under this approach are, for example, Shen et al. [113] where the authors exploit implicit feedback information to select the appropriate terms from the preceding query and its corresponding search results, in order to expand the current query. Chirita et al. [26] propose five different techniques for generating the expansion terms, by analysing user data at increasing granularity levels and using external thesaurus. They also propose to adapt the expansion process to different features of each query, such as the query clarity [31]. Zhou et al. [141] propose a query expansion framework based on user profiles mined from the user social media data, such as tags and marked resources. Current and more focused approaches are [29] and [52]. In the former, Craveiro et al. show a temporal query expansion, where the documents text is temporally segmented to create a relationship between words and dates, particularly useful for time-sensitive queries. In the latter reference, Hahm et al. propose a personalized query expansion approach for engineering document retrieval, where both user interests from the user profile and intent from the user task context are used to expand the query.

2.3.2 Within the search

There are some articles which modify the search engine retrieval model in order to account for personalization using the user profile information, being most of them

focused on link analysis, and specially on PageRank [90]. However, this approach has not been as commonly used as before and after the search approaches [117], mainly because within these two approaches the modifications needed to be done are usually easier than those needed in order to modify the retrieval model, and sometimes the access to the retrieval model is not feasible. However, to include personalization within the search itself, i.e. modifying the retrieval or ranking model, is a very suitable way for IR personalization. It avoids to include extra information to personalize the search (before the search), or to reorder the retrieved results a posteriori (after the search), but the retrieval model itself includes the personalization features, and the obtained results just after the search are already personalized.

Focusing on the PageRank retrieval model adaptations, Haveliwala [54] 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 and then combined at run-time, based on the similarity between the user query and each of the 16 topics. In order to generate topic oriented rankings, Nie et al. [87] distributed the PageRank of a page across the topics it contains, using a random walk model that probabilistically combines page topic distribution and link structure. Jeh et al. [63] present new graph-theoretical results modifying the basic PageRank algorithm to create personalized views of the web, which redefine the importance considering the user profile information. This modification of the algorithm encodes personalized views of importance for more refined searches as partial vectors, being these partial vectors shared across multiple personalized views. Bahmani et al. [8] show a fast MapReduce algorithm for Monte Carlo approximation of personalized PageRank vectors of all the nodes in a graph, doing very efficient single random walks.

There are some other approaches not focused on PageRank, such as: Chang et al. [19], where the authors modify the HITS algorithm [67] and present a technique for learning a user internal model of authority manipulating the weighting of the link matrix. Outside the field of link analysis, Teevan et al. [127] modified the probabilistic ranking function BM25, which ranks documents based on their probability of relevance given a query, by weighting the terms appearing in the user profile higher. More recently, Wang et al. [135] propose a general ranking model adaptation framework, by first training offline a global user-independent general ranking

model. This general model is then adapted to better fit each individual user preferences, by applying a set of linear transformations, such as scaling and shifting, over the parameters of the given global ranking model. Not only ranking accuracy, but efficiency is also a primary consideration in this work. The authors then applied their general framework to three popular ranking models: RankSVM, RankNet and LambdaRank. Lastly, Song et al. [117] use a deep learning approach for personalized ranking adaptation. They first train a deep RankNet as a general user-independent ranking model, and then they adapt this global model to each individual user, by training individual models based on the search history of the user.

2.3.3 After the search

The most common personalization technique after the search is a reranking process. It tries to improve the user satisfaction reordering the original query top retrieved results, taking into account the user profile information. As users normally do not go beyond the second page of results, according to Spink et al. [121], and due to performance requirements, the reranking process usually only consider the top retrieved results.

The main advantage of this approach is an increase on the final reordered result list precision, which is one of the most desirable and observable feature for users. This higher precision is specially appreciated in web environments, where users do not generally check too much retrieved results. This characteristic is less important in professional environments, where recall is also an important factor. Another advantage is that the whole personalization process may be performed on the client side, without the need to send any personal data to the server. This feature is important for privacy concerns but also for scalability, since the server simply executes the user query without any other overhead, being the load of this process assumed by client machines. Additionally, if the reranking process is carried out in the client machine, more results may be considered into this reordering process (the more the better), than if it was done in the server. On the other hand, and as the most important disadvantage, a reranking process only allows to reorder the already not personalized IRS retrieved results. Since the reranking process is performed after the search, it only allows to improve the original results, but not to include any

new result which may be of interest for the user considering his/her user profile information.

There are plenty of studies following this strategy. For example, Sugiyama et al. [123] use a keyword-based user profile and rerank the results based on the similarity between each web page and the user profile. Shen et al. [113] use implicit feedback information to exploit the viewed document summaries to rerank the user not yet seen documents. Chirita et al. [25] focus on reranking the web search output according to the cosine distance between each page and a set of Desktop terms describing user interests. Teevan et al. [127] personalize web search by using an automatically constructed user profile based on previously issued queries, visited web pages and documents and emails created or read by the user. This user profile information is used to rerank web search results within a relevance feedback framework. Lastly, Matthijs et al. [78] build a user profile considering the user complete browsing behaviour. Then, they use this information to rerank web results, by using a combination of content and previously visited websites.

In a context different from personalization, namely that of methods for fusion of retrieved lists, Meister et al. [79, 80] rerank a list retrieved in response to a query, using a second list retrieved by the use of a different retrieval model and/or query representation. They exploit inter-document similarities between the lists in order 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 [79]. Similarly, Zighelnic et al. [142] fuse the results retrieved in response to the original query and to its expanded form. This contributes to alleviate the query-drift problem derived from the used expanded query. Using both lists of results, they get significantly better performance than that of the retrieval based only on the original query, and more robust results than those of the retrieval only using the expanded query.

2.4 Evaluation

Evaluation is the process that measures how good any system does the task it was designed for. Traditional IRSs have an underlying retrieval model which tries to return a ranked set of relevant results, based on a given query. To do that task,

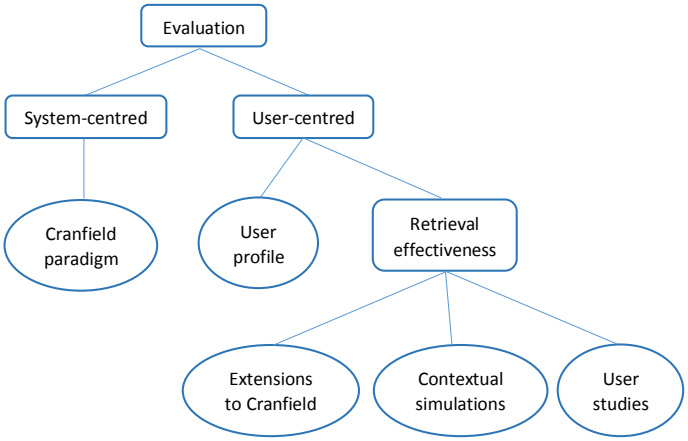

Figure 2.3: Classification of the main IR evaluation approaches.

the retrieval model needs a representation of the documents and the query itself. The retrieval model tries to match both representations to select and rank those documents which better satisfy the user information needs, represented by the issued query. These systems are considered as system-centred approaches. However, personalized IRSs are those in which, besides to the issued query, additional information about the user context is considered in the retrieval process. These IRSs are considered as user-centred approaches.

We next show a literature review and make a summary of the evaluation characteristics of system-centred and user-centred approaches (see Figure 2.3 for a graphic glance), focusing more on the latter ones, which is the approach this thesis is focused on.

2.4.1 Evaluation of system-centred IRSs

The general followed process to evaluate the retrieval effectiveness of a system-centred IRS may be summarized as follows: for a given query, the retrieval model which is going to be evaluated returns a ranking of results ordered by decreasing relevance degree. A similarity measure is then calculated, comparing these results against the set of results which have been previously manually assigned as relevant (relevance assessments) for this given query. The higher the similarity the better is the retrieval effectiveness of the evaluated model.

Historically, to evaluate system-centred IRSs, an evaluation methodology based on the Cranfield paradigm proposed by Cleverdon et al. [28] is used. The evaluation framework consists of a test collection composed by a document collection, a set of well-defined queries and a set of manually assigned relevance assessments for each query. The main evaluation metrics of this paradigm are the well-known precision and recall, and some others generally based on them. The manually assigned relevance assessments are used by those metrics, in order to get useful evaluation values to compare the retrieval effectiveness of the different models or systems. A more technical view of the evaluation process, including different evaluation metrics and more, will be given in Section 3.6.

The Cranfield laboratory-based evaluation framework has been used for many years in the IR area. It has a number of advantages that have allowed the IR continuous development, obtaining better retrieval results each time. Some of these advantages stand out: 1) it allows *repeatable* and *comparable* evaluation experiments. The ability to repeat an experiment is considered as a key characteristic of any empirical study, according to Ramesh et al. [100]. Systems which are not able to be evaluated under this kind of evaluation frameworks have big problems in their development and improvement. For example, the evaluation results obtained from the inclusion of a possible system improvement will not be comparable with previous results, since both of them will have been obtained under different circumstances, making impossible to discern whether the improvement is real or not. And 2), under the same experimental conditions the findings are *generalizable*. If relevance assessments are large enough, test collections are reusable, according to Zobel [143]. This characteristic allows new systems and models to be evaluated with the same documents and queries, applying the same relevance assessments.

Traditional evaluation frameworks designed to evaluate system-centred IRSs are not suitable to evaluate the new cognitive side derived from the introduction of the user in the retrieval process. Both, the retrieval and the evaluation, ignore the influence of the previous user cognitive side in the whole retrieval process [59]. Some examples of the previous affirmation are the following: system-centred IRSs assume that user queries are a good representation of the user, which leads to serious problems on the reliability of real life relevance assessments [131]. Similarly, system-centred IRSs assume that user queries are well defined, being a good representation

of this user information needs, while they actually are almost always ambiguous. Therefore, the extracted conclusions are not always truly generalizable [12].

2.4.2 Evaluation of user-centred IRSs

As it may be derived from the previous section, an alternative to the Cranfield evaluation laboratory-based model is needed for evaluating user-centred IRSs. There are several evaluation frameworks, but not any standard, for user-centred systems. For this reason, we next give an overview of the most important evaluation strategies, being mainly guided by the broad survey done by Tamine et al. in [125].

Considering the new characteristics within a user-centred evaluation approach, the evaluation process could be separated in two main different steps: the user profile evaluation step and the retrieval process evaluation step.

In the *user profile evaluation step*, the main objective is to measure the user profile accuracy. User profiles are viewed as the representation of user models. There are three main user profile representations in order to store the information about the user: weighted keywords, semantic networks and weighted concepts. The last two, sometimes enhanced with the use of thesaurus or ontologies. The question is: to what extent the user profile representation, which models the user, is a reliable portrait of this user? As usual, there are no standard evaluation metrics to answer the previous question. Additionally, these evaluation metrics are weakly dependent on the selected user profile representation. For example, Liu et al. [73] use a set of concepts issued from a reference ontology as the representation for the user profile. To measure the user profile accuracy, they map the queries against the user profile to identify the most related categories. These related rankings are then compared with the categories manually assigned by the users for the same queries. Similarly, Ding et al. [44], also use a set of concepts from a reference ontology, stored as a self-organizing map using a neural network classifier algorithm. They use documents annotated with related categories from the ODP ontology to train the classifier. To test the classifier and measure the user profile accuracy, some test documents and new user queries with their related categories are used. These categories are then compared with those manually annotated.

A good representation of the user profile, in terms of having the biggest accuracy representation with respect to the real user interests and preferences, is very

important. The quality of personalized results will highly depend on the user profile quality and how it is exploited in the retrieval process.

Once the user profile is built, its information will be used by the implemented personalization techniques, in order to satisfy the user information needs, while also obtaining the closest results to the user interests and preferences. The personalization technique to be tested, together with the user profile, will have quite a few configuration parameters to be adjusted. The final evaluation goal is to find the joint configuration which maximizes the personalized IRS retrieval effectiveness.

There are three main evaluation frameworks to measure the *retrieval effectiveness* of user-centred IRSs:

Extensions to the Cranfield laboratory-based evaluation. These extensions were the first attempt to perform a more user-centred evaluation framework. They model a small interaction between the system and the user, including some metadata about the user (e.g. genre or location) and the query (e.g. purpose). With the inclusion of real users, they try to make the evaluation process more realistic and relatively controlled. TREC interactive track [53] and HARD track [3] are examples of this kind of evaluation frameworks, which mainly compare a baseline run ignoring the user/query metadata with another run considering it.

The following are the most common extensions to the laboratory-based evaluation characteristics: the document collection is usually provided by a controlled IR framework, such as TREC. The topics are based on the controlled IR framework, but enriched with annotated metadata. The relevance assessments are either obtained from the controlled IR framework or may be provided by users. Accordingly, the users may be the controlled IR framework assessors or real searchers. The usual evaluation metrics are precision, recall, F-measure or user effort. The evaluation is performed by a comparison between a run only involving the query and a final run with the query including its associated metadata, or in interactive systems by comparing their interactive behaviour against the fully automatic version.

However, these extensions to the laboratory-based evaluation framework are still system controlled, their evaluation metrics are still based on precision and recall, and their capability of capturing the contextual aspects is very limited in all the process, allowing a restricted personalized evaluation.

Contextual simulations. They simulate users and user-system interactions through a well defined retrieval scenario. They are also called hypothesis-based evaluation studies by Petrelli in [98], since she call hypothesis to the well defined retrieval scenarios. A scenario represents possible user-system interactions, which the retrieval model should consider to provide better results to the given simulated user.

Contextual simulations have been proposed to attenuate the limitations of laboratory-based evaluation extensions. Instead of modeling a minimal interaction, contextual simulations are able to model different user interactions, used retrieval strategies, and external factors, which could influence in the user interaction decisions.

Thanks to the ability of simulating user-system scenarios, contextual simulations are used for the following two main purposes:
The development of IRSs measuring the contextual retrieval effectiveness. In [115], Sieg et al. present a personalized search system which builds models of user context as ontological profiles. These user profiles are built assigning user interest scores, derived in an implicit way, to concepts from the ODP ontology. Since the user behaviour changes dynamically, a spreading activation algorithm is used to maintain these interests updated. They show that reranking the search results, based on the ontological based user profile, helps to present the most relevant results to the user.

The development of the associated interfaces design. In [137], White et al. develop search interfaces and search scenarios, which interact with different retrieved information such as the title, the summary or a sentence in context, with the objective to test several implicit feedback models.

The following are the most common contextual simulation characteristics: the document collection is usually provided by a controlled IR framework such as TREC, or it is based on a set of online open source web pages. The topics depend on the document collection, and therefore, they may be provided by the controlled framework, or they may be even automatically generated without any user involvement as in [115]. The relevance assessments are given by the controlled framework, or depend on whether the document is classified or belongs to the concept or user interest being simulated. Users are simulated by hypothetical context situations. Usual evaluation metrics are precision and recall at cut-off n, and the standard Mean Average Precision (MAP). Finally, the evaluation is performed without the interaction of real

users, and performing a comparison between a run only involving the query and a final run with the query personalized using the context.

Contextual simulations are worthwhile since they are less resources consuming than experiments with real users, and they provide comparable evaluation results within the same retrieval scenarios. However, they also have some disadvantages, such as the topics and collection might not be interesting to many searchers, the effort to define the retrieval scenarios or hypotheses is usually hard, and the relevance assessments are fixed by assessors, which probably have a very different background knowledge with respect to the real users.

User studies. They are the best evaluation method from the qualitative point of view, since the IRS effectiveness is directly evaluated by real users in a real retrieval environment. User studies include all user interactions with the IRS and several ways of feedback, such as questionnaires, interviews, as well as a constant monitoring of the user behaviour with respect to the preassigned search task within the IRS.

User studies are commonly performed by simulating work situations [14, 107], which pursue to involve individual users into a similar environment and information need search tasks, with respect to their daily job. For this reason, the search task must be appropriate for the user related to his/her experience with the given task. Users range from expert users (project team members, technical employees,...) to common users (normal citizens, children,...).

This evaluation approach also has some disadvantages, such as its enormous time and resources requirements, which limit its realization most of the times. In order to perform a user study, real users must be involved. We must think that all the work a user must do in a user study is very hard and time consuming. They normally have to spend several hours performing searches, reading a lot of information, checking the relevant documents found, etc., and when they finish, they usually have to fill one or more questionnaires. There is another inconvenient if the user study involves expert users. They usually are busy people, which not only limit their availability, but also keep them out of their jobs, meaning a lose of money for them and/or their companies or institutions. Additionally, sometimes a physical place and computational resources are also required to accommodate the users during the user study.

Performing a user study is very difficult due to all of the previous characteristics, but there is another issue to take into account: the balance between control and realism. The experiments are not repeatable if different users are involved in different user studies, or even if not always the same users are involved in the same tasks, because of the individual differences between them, such as their background knowledge about the task, their intelligence or their familiarity with the search interface, for example. The previous fact makes very difficult to discern the influence of the system evaluated variables over the overall retrieval effectiveness, making the conclusions not generalizable. To try to diminish the previous problems, several recommendations are given in order to ensure a minimal reliability of user study experimental findings in [12, 24]: the more users the better, minimal interactions between users, these must ignore which system is being evaluated, permute order of search tasks between users, run a pilot study before the main study, etc.

The following are the most common user study characteristics: the web conforms the document collection, being consulted through a public search engine. Topics are created, selected from a predefined set, or both, by the users. Relevance assessments are based on the user click-through data or explicitly made by them. Almost anybody could be a user but, most of the times, they have some knowledge about the IRS or the collection. Some evaluation metrics used in user studies are MAP, precision at cut-off n or NDCG (Normalized Discounted Cumulative Gain) [62]. In the user study evaluation protocol, the user interacts with the system performing search tasks in different domains, performs the topics as stated above, and judges relevant documents among the first results list. All this information is stored in a log file for performing the evaluation. This log file usually consists of the user queries, retrieved results, clicked results or relevance assessments, and any other important contextual information.

Both system-centred and user-centred IRSs evaluation frameworks (especially the user studies in the latter case), are the two poles of the evaluation range. Both of them have merits that could be exploited at different stages of the IRS design [98]. In order to guarantee the main IRS technical objectives, system-centred evaluation is more suitable in their first design stages, allowing to have controlled and repeatable experiments. However, user-centred approaches are more suitable for further stages, since they introduce the search context into the IRS design, allowing the

evaluation of the system improvement from the user dynamic perspective. Another claim about the advantages of using both approaches along the different stages of any IRS is done in [43], where Díaz et al. affirm that the use of both approaches offers more information about the system real performance than any of them separately.

In this chapter we have presented a formal overview of the whole personalization process. We have introduced some concepts and show how the use of IRSs is very important to find relevant information for most users nowadays. We also have shown why because of different factors (mainly because of the exponential information overload we are faced with everyday), personalization has become almost necessary to improve the retrieval effectiveness of traditional IRSs. After the introduction, the rest of the chapter explains the different personalization process steps, including some references with the main research contributions to the field. The first personalization step is about how to build the user profile, starting on how to gather the user information, how to represent it in a user model, and finally how to keep this information accurate and updated. Once we have the user profile, its information is exploited by personalization techniques, which we have classified depending on where the user profile information is used: before, within, or after the search has been performed. The final personalization step is how to evaluate the whole process. We have shown why the traditional system-centred evaluation frameworks are not suitable for the evaluation of personalized IRSs, and explained three different personalized evaluation frameworks for the correct evaluation of user-centred IRSs: extensions to the Cranfield laboratory-based paradigm, contextual simulations, and user studies.

Chapter 3

Structured Information Retrieval

3.1 Introduction

Documents are basically composed by a set of terms, describing the content, which is organized around a well defined structure. This structure, e.g. chapters, sections, paragraphs, etc., makes the document content more readable and comprehensible for readers. Traditional IR only focus on the text of the documents, not considering their structure at all, viewing them as a bag of words (plain documents). Therefore, the minimal retrievable result for traditional IR is the whole document, independently of the document portion of text actually relevant for satisfying the user information need. However, under Structured IR documents are not considered anymore as atomic units of information to be retrieved only as a whole, but only the real relevant parts of the document will be retrieved (also the whole document if applicable). Obviously, this fact is very beneficial for users, since they are taken straight to the document relevant parts, implying a save of time and effort for them, especially when dealing with large documents.

The standard language to represent and exchange structured data is XML (eXtensible Markup Language), to the point that structured IR is also known as XML-IR. 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 in a hierarchical way, instead of atomic entities. Traditional IR is not able to exploit this structured characteristic to carry out a more focused retrieval.

In fact, the main XML-IR asset [69] is to take advantage of the document internal structure, allowing to retrieve both specific parts of the documents (Structural Units (SUs)) as well as complete documents. This will depend on the user needs and the distribution of the relevant information, across the different parts of the XML document. The most appropriate SU to be retrieved is a quite difficult election, where the *structured document retrieval principle* [76] must be followed: "a system should always retrieve the most specific part of a document answering the query".

It should be noted that this thesis always refers to XML documents as a container for text (text-centric XML view), and not to XML documents as a container for data (data-centric XML view). The latter are more suitable for database-style searches, where the user is interested in exact matches, not being appropriate for IR.

These new structural characteristics require new designs and/or adaptations of traditional IR techniques and evaluation metrics. They cannot simply be reused under this new approach, basically because of the dependency between XML document components. This document component dependency causes the following two main XML intrinsic difficulties [65]: (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, e.g., 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 extra difficulties in terms of design and evaluation.

The main workflow of a traditional IRS, see Figure 3.1, is the following: the *documents* are the source where all the information resides. These documents must be transformed into an efficient and easily accessible data structure (*index*) on which to perform the searches. Then, users are able to send *queries* to the IRS to satisfy their information needs. These queries will be matched (*retrieval model*) against the previous built index, and a list of ranked results will be returned and presented to the user by the IRS. In addition to the previous stages, there is an extra and important stage, the evaluation step. This process is very important to measure the IRS performance, allowing to continuously test and improve it. All the previous stages must be adapted to work with structured IR.

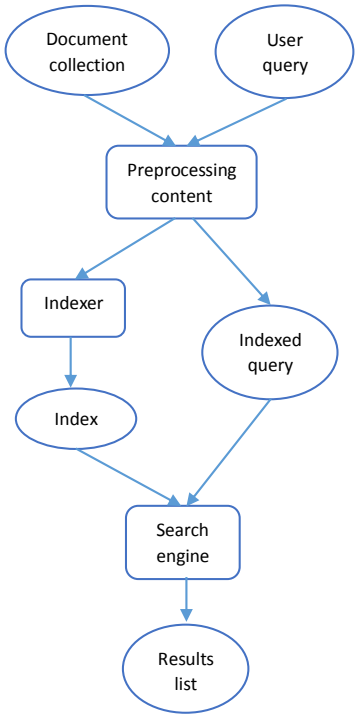

Figure 3.1: A basic traditional IRS workflow.

The rest of the chapter is devoted to briefly explain the IRS (traditional and structured) internal functioning and stages, in order to better understand the improvements and contributions made in this thesis; see Part III. The last section of the chapter, 3.7, describes Garnata [41], a structured IRS based on probabilistic graphical models developed within our research group, which is used most of the times in this thesis experiments.

3.2 Documents representation and the indexing process

A (text) *document* is basically a set of terms with some meaning, separated by punctuation marks and stored in a file. Since the search over the file system document files, considering the document text as it is stored on those files, would be totally inefficient, this text must be transformed into a more searchable efficient data structure, the *index*. In this section, we explain this process for both plain (unstructured)

and structured documents[1].

From a semantic point of view, only a part of the document content is useful for searching [7]. Consequently, not all the document content is included in the index. Before the index build step, some preliminary transformations are performed over the original document content (only the text for structured documents). These transformations allow some advantages, for example, the physical space required to store the final index is considerably lower.

The following are the most common transformations performed over the documents content:

- *Tokenization*: this process breaks up a character sequence into pieces, called *tokens*, usually removing at the same time the punctuation marks. A token is a sequence of characters grouped together as a useful semantic unit for processing. The tokenization process is usually carried out using the whitespace character as delimiter, but there are a number of difficult and tricky cases, being a language-dependent process.

- *Deletion/transformation of non alphabetic/diacritical characters*: deletion of punctuation marks (if not already done in the previous step), numbers (commonly, but not always done), or any strange character. Diacritical characters are transformed into their corresponding non-diacritical character.

- *Case folding*: all characters are converted into their lowercase version.

- *Stopwords deletion*: a *stopword* is an extremely common term, which almost does not contribute to the sentence semantic meaning. They usually are prepositions, articles, etc. The following are some stopword examples: the, a, at, by, for, that, etc. This step contributes to highly decrease the size of the final stored index, since they are very frequent. However, this step is not recommended if the IRS supports phrase queries, normally indicated with this part of the query text surrounded by double quotes. Obviously, stopwords are also language-dependent, existing predefined stopword lists for different languages.

[1]Whenever we mention the word *document*, we are referring to its textual content and/or structure, and not to the digital document itself.

- *Stemming*: this process removes the most common morphological and inflectional endings from words, i.e., it converts every term into its corresponding root, since all terms with the same root may be considered to have the same meaning. Two different examples are: it always gets the singular term forms, and discards all the verb derivations getting its root form. This is again a language-dependent step, also contributing to reduce the final index size. The most common and a empirically proven effective stemming approach is the *Porter's stemmer* algorithm [99][2]. It basically consists of different sequentially applied phases of word reductions, by applying a set of rules in each phase.

The final terms version after completing all the previous stages are the included terms in the index. This index will be the efficient and easily accessible data structure where the searches will be performed. An index build process is very complex and it is beyond the scope of this thesis, so we are going to explain the basics of the most common index data structure, the *inverted files*.

An inverted file is a data structure with two main components: *vocabulary* and *occurrences*. The vocabulary is the set of different terms appearing in the whole collection of documents, and occurrences are the places (document identifiers, and sometimes positions within those documents) where these terms appear in the text. Each vocabulary term has a pointer to its occurrences list.

For unstructured documents, the indexing process to build a rather simple inverted file is as follows: each document in the collection has a unique identifier (*docId*). The input to the index build process is the list of terms after having gone through all the previous preprocessing steps. For each of those terms, if it was not already present in the vocabulary, a new entry for this term is created in the vocabulary and its corresponding document *docId* is stored as part of its occurrences list, or only the last step if the term was already in the vocabulary and it was not previously been found in the current document. This process can get complicated if we also want to store each term position within each document, or any additional feature. Both, the vocabulary and the occurrences lists are alphabetically sorted to increase the query processing efficiency. The vocabulary also stores some statistics,

[2]Different language stemmer versions can be found in http://snowball.tartarus.org/texts/stemmersoverview.html

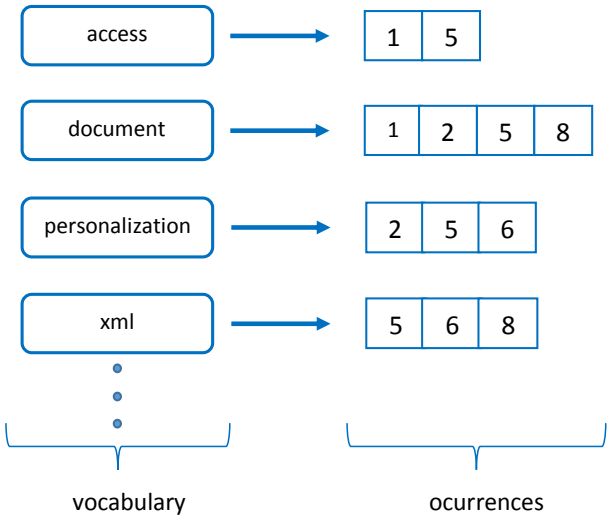

Figure 3.2: An inverted file example.

such as the term and document frequencies (the total term number of appearances in the whole collection, and the number of documents each term appears in, respectively), which improve the search efficiency at query time, being also used in many ranked IR models. Figure 3.2 shows an example of an inverted file.

As the reader may imagine, the size of the vocabulary is rather small (lower than 40MB for big collections [139]), in comparison with the size of the occurrences (normally between 30% and 40% of the original text size). For this reason, vocabulary and occurrences are usually stored in different files, being the former normally kept in memory and the latter in disk.

With respect to ***structured documents***, both the document representation and the indexing process are a bit more complicated.

As we have already mentioned, we are going to use XML documents as the representation format for structured documents. An XML document is basically an ordered and labelled tree. Each node of the tree is an *XML element*, which is represented with an opening and closing *tag*. These tags show the boundaries of the XML element. Each of these elements may have one or more *XML attributes* in the opening tag. The text between the opening and closing tags is the content of this XML element. Inside an XML element may exist other XML elements following a

```
<article>
   <author>Eduardo Vicente</author>
   <title>Personalizing XML</title>
   <chapter number="1">
      <section number="1">
         <subsection>IR is finding...</subsection>
         <subsection>Personalized search...</subsection>
      </section>
   </chapter>
</article>
```

Figure 3.3: A fragment of a sample XML document.

hierarchical structure. We can create all the XML elements we need for representing the documents content and structure. For example, in the Figure 3.3 XML sample, the *section* element is defined by the opening and closing tags *<section ...>* and *</section>*. This element also has an attribute *number* with value *1*, and two child *subsection* elements.

Figure 3.4 shows the article in Figure 3.3 represented as a tree. The leaf nodes of the tree contain the content (text), e.g. the *subsection* element with the text *"IR is finding..."*, and the internal nodes represent the structure of the document. The standard for accessing and processing XML documents is DOM (Document Object Model)[3]. DOM is a platform and language-neutral interface that allows to dynamically access and update the content, structure and style of XML documents. This process starts from the root element descending down the tree from parents to children.

XPath (XML Path Language)[4] is a standard language to find and process any XML element within an XML document. An XPath sequence represents the path of an XML element following the tree structure, usually describing each element in the path by its tag name and its position within the tree. An XPath example of Figure 3.4 is */article[1]/chapter[1]/section[1]/subsection[2]*, referring to the second *subsection* element with the text *"Personalized search..."*.

[3] http://www.w3.org/DOM/

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

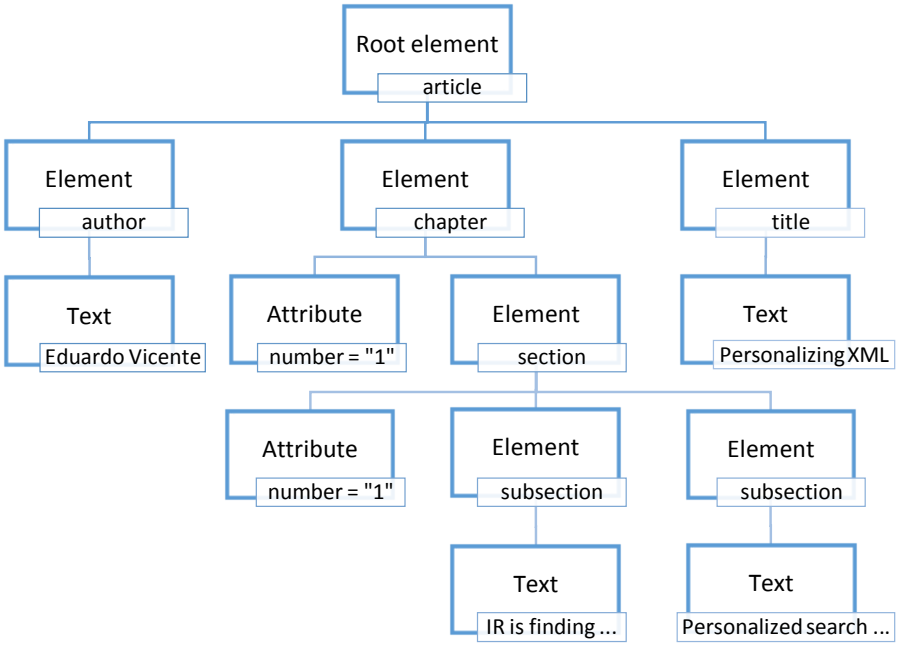

Figure 3.4: The XML document from Figure 3.3 as a tree.

When we deal with XML documents, under several situations we must know the structure of the documents. For this reason, the XML documents must follow a series of constraints to be considered as a valid XML document, for a specific purpose or system. These constraints are defined in the XML *schema*. A schema for *articles* in Figure 3.3 could specify that *section* can only appear as a child of *chapter*, and that only these both elements can have the *number* attribute. Two schema standards for XML are *DTD* (Document Type Definition) and *XML Schema*. The former is the one used in this thesis.

Concerning to the indexing process under XML retrieval, it also appears a new difficulty. The retrievable units are not predefined, as with unstructured retrieval, where the retrievable unit is the whole document. In XML retrieval, the whole document, a section or even a paragraph may be potential answers to a query. Therefore, the determination of the indexing unit is an important decision in structured IRSs, because indexing all of them is neither necessary nor efficient most of the times.

There are different indexing strategies for XML IR, well explained in [69]. We are going to give a rough outline of some of them:

- *Element-based indexing*: It is the simplest approach, since it indexes all the elements of the XML document. Each element is indexed based on its own and all its descendants text.

- *Leaf-only indexing*: Since this strategy only indexes the leaf elements, the resulting index size is much smaller than with the previous strategy. This index only allows to estimate the relevance of leaf elements, requiring a propagation mechanism to compute the ancestors relevance in the hierarchy.

- *Aggregation-based indexing*: This strategy aggregates the statistics of each element with the statistics of all its descendants.

The IRS most of the times employed in this thesis uses the leaf-only indexing approach. Since this IRS is based on Bayesian networks, it naturally performs relevance propagations, while allowing to have a more reduced index structure.

3.3 Queries

Once we have already seen how documents are represented and indexed into an efficient searchable structure called index, the next step is to allow users to send requests to the IRS. When a user faces an IRS wants to solve some information needs. The way to communicate these information needs to the IRS is through the use of *queries*.

The intention behind the user issued query is to retrieve the best possible indexed results, which satisfy the user information needs. To do that, the IRS matches the query against the index. For that reason, it is important to use the same preprocessing steps (some of them language-dependent and with some tricky stages) for the query, as those used in the indexation process, in order to obtain more reliable matches between the query and the index.

Considering non-structured and structured IR, there are two different types of queries, CO (Content-only) and CAS (Content-and-Structure) queries.

Content-only queries (CO). A content-only query, as its own name suggests, is a query expressed in natural language, where there is only information about the

content the user wants to retrieve. This type of queries are broadly used in most current IRS, including the very popular web-based ones.

These queries are used in traditional IRSs, but they can also be used in structured IRSs. The only difference will be the list of retrieved results. For traditional IRSs, the results will be the matching whole documents, while for structured IRSs, the results may be either the whole documents or any retrievable structural unit (XML element), e.g. a chapter or a paragraph.

Examples of this query type could be: "olive oil production" or "research in andalusian universities".

Content-and-Structure queries (CAS). A content-and-structure query, as its own name suggests, is a query where there is information about *what* the user wants to retrieve, and *where* this information should be located. The *what* involves the specification of the content, while the *where* is related to the structure of the documents. To express these new structural requirements is no longer enough to write the query as a set of keywords in natural language.

There are some state-of-the-art querying languages for structured IR, such as XQuery[5] (supported by XPath) or NEXI (Narrowed Extended XPath I) [130], widely used in INEX (INitiative for the Evaluation of XML retrieval)[6]. But they have two key disadvantages: 1) they are complex to learn how to use them, and 2) users must know the structure of the documents (schema), which most of the times is not the case.

Perhaps for these reasons, although there are many IRSs able to deal with XML documents[7], often these systems only process CO queries. These query languages are more suitable for expert users, letting them to specify these kind of structural units that will much better satisfy their information needs, in opposition to the classic keyword search. Nevertheless, a general method used to convert some of these only CO-able systems into fully structured IRSs, which can process CAS queries, has recently been proposed by de Campos et al. in [37].

Concretely, in this thesis we use NEXI for structured queries. To better understand how NEXI queries are formed, we firstly start explaining the basics of XPath,

[5] http://www.w3.org/standards/techs/xquery

[6] https://inex.mmci.uni-saarland.de/

[7] The series of INEX Workshop proceedings are an excellent source of information.

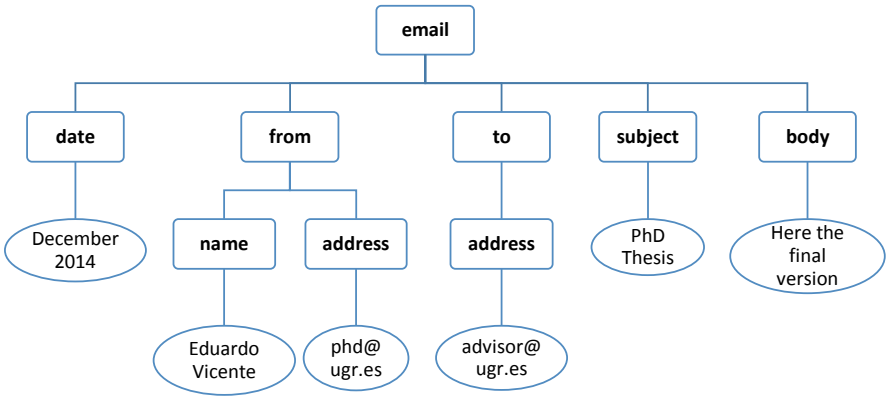

Figure 3.5: An XML document as a tree.

guided by the XML document, represented as a tree, in Figure 3.5.

The name of an element, (A), selects all elements with that name (e.g. *address* selects two nodes of the tree). A slash '/' is used to select child nodes in the tree (A/B, where B is a direct descendant of A – *from/address*). A double slash '//' means that any number of elements could be included in the path ($A//B$, where B is a descendant, though not necessarily a direct one of A – *email//address* selects *address* units directly or indirectly contained in an *email* element – *email/from/address* and *email/to/address*). A slash at the beginning of the expression means that the path starts at the root element ($/A$). An asterisk '*' selects all the elements placed in the path after it ($/A//*$ – */email//** selects all the descendants of *email*). Finally, a pair of opening and closing brackets, with a number between them, after an element establishes the order of the element as a child from left to right ($//A/B[3]$ selects the third B element child of A).

NEXI is a small XPath subset with an additional *about()* clause. This 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. This *about* clause is used for identifying elements about any given topic.

The general form of a NEXI CAS query is $//A[B]//C[D]$: "returns C descendants of A, where A fulfils the condition B and C fulfils 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.

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 (*what*) and the *book* units are the context (*where*).

3.4 Information retrieval models

Now we have already seen how documents are indexed and how users can send queries to the IRS, the next step is to explain the way both items are matched, in order to retrieve a list of results to satisfy the user information needs. An IRS must implement a given *retrieval model*, which performs this task.

The retrieval model computes the importance of terms in the query and documents, i.e. their *similarity*, to determine the IRS output for a given query. The similarity $sim(q, d)$ between the query q and the document d is usually defined as:

$$sim(q, d) = \sum_{t \in q} w_{t,q} \cdot w_{t,d}, \tag{3.1}$$

where $w_{t,q}$ and $w_{t,d}$ are the weights of term t in query q and document d, respectively, according to the system *weighting scheme*. These weights are assigned in order to allow a ranked retrieval, which is one of the best IR contributions, especially when dealing with large document collections.

There are plenty of weighting schemes in IR [76, 92], but we are going to explain maybe the most famous and easy to understand, the *tf-idf* weighting scheme. The first idea is: the more often a query term q_t appears in a document d, the more

related will be q and d. This weighting scheme is known as *term frequency*, and it is denoted as $tf_{t,d}$. But clearly, all terms in a document are not equally important for assessing relevance. For instance, if a term is very frequent in a document collection, this term will not be very useful to discern the relevance of documents, since it will appear in almost all of them. To avoid this problem, the idea is to reduce the *tf* weight of a term t, by a factor that grows with the number of documents in the collection where t appears, known as *document frequency* and denoted by df_t. If we denote N as the total number of documents in the collection, we can define the *inverse document frequency* (idf) of a term t as:

$$idf_t = log\frac{N}{df_t}.$$

According to the previous equation, *idf* values will be high for strange terms and likely low for frequent terms. Combining the two previous definitions, we may compute a final weight for each term t in each document d, following the *tf-idf* weighting scheme as follows:

$$tf\text{-}idf_{t,d} = tf_{t,d} \cdot idf_t. \tag{3.2}$$

Hence, considering the previous Equation 3.2, Equation 3.1 may we rewritten as:

$$sim(q,d) = \sum_{t \in q} w_{t,q} \cdot tf\text{-}idf_{t,d}, \tag{3.3}$$

where $w_{t,d}$ has been substituted by $tf\text{-}idf_{t,d}$ as the used weighting scheme.

The task any IR model must do is to compute the function $sim(q,d)$ for each document in the collection, and return a list of results to the user, accordingly to the computed values. Next, we are going to explain three classic IR models.

Boolean model. The Boolean model was the first used IR model approach. This retrieval model is based on boolean logic and classic set theory, thus, not following this section introduction explanation and equations. Within this model both the query and documents are considered as sets of terms. Queries are represented as boolean expressions (terms joined by *and, or, and negation operators*). The retrieval is based on whether or not the documents contain the query terms, using the different operations to work with sets (union, intersection, and complementary set). The

final list of retrieved results will be composed by those documents, which verify the query boolean expression following a binary decision; i.e., given the query q, for each document d, this model similarity value $sim(q, d)$ will be one (d will be retrieved) or zero (d will not be retrieved).

This model advantages are the following: it is very simple to understand and to implement, being at the same time very efficient in performance. However, it has several disadvantages, such as: it provides an unranked output results list (all documents are considered equally important, when obviously they are not), it is more like a data-oriented style search rather than IR, for users it is harder to write boolean than natural language queries, and due to the exact matching this model may retrieve too few or too many documents.

Vector space model. This IR model, presented by Salton et al. in [105], considers each collection document as a vector, $\vec{v}(d)$, in the common vector space formed by the indexed vocabulary terms. These terms have an associated weight, given by the Equation 3.2 (or any other weighting scheme), if the term appears in the document, or equal to zero, otherwise. The query may be considered as a very short document, therefore, it may also be seen as a vector, $\vec{v}(q)$, in the same document vector space. This feature allows to calculate the *similarity* between a document d and a query q computing the cosine of the angle between their normalized vector representations, $\vec{v}(d)$ and $\vec{v}(q)$, respectively. The higher the cosine similarity the more relevant will be d with respect to q; see [76] for a deeper explanation of this model.

This model advantages over the previous boolean model are the following: it allows partial matching with a continuous similarity degree between queries and documents, which in turn allows the retrieval of a ranked list of results, while still being a simple model based on linear algebra. It also has some disadvantages, such as: large documents are poorly represented with poor similarity values, and the order of terms in the document is lost.

Probabilistic model. This IR model was presented by Robertson and Sparck-Jones in [101], being [118, 119] from the same authors, a comprehensive presentation of the model with several comparative experiments, which demonstrate the model effectiveness and robustness. It tries to estimate the probability that a user finds a document d relevant for a query q, assuming that this probability only depends on the query and document representations. It also assumes that there is a subset

of collection documents preferred by the user as the relevant results for the query. This subset should maximize the overall probability of relevance for that user, being documents within this subset relevant, while documents outside this subset are non-relevant to the query. It should be noted that the well-known *BM25* ranking function [102] is based on this probabilistic model.

The main advantage of this model is that results are sorted by decreasing order of relevance probability. Some disadvantages are that an estimation is needed for the initial run probabilities, and terms are assumed to be independent.

Concerning ***structured IR models***, the used retrieval model depends on the choice of the indexing approach. Most structured retrieval models are adaptations of the non-structured IR models. These adaptations try to exploit the additional structural information contained in XML documents, although most part of the relevance of an element can be estimated only based on its content. Concretely, for CAS queries, some additional processing is necessary to comply with the query content and structural information.

A good reference to check structured IR models is [69]. We next provide a very brief overview of some of them.

Element scoring. This model tries to estimate the relevance of an element, only based on the information provided by this element. The scoring function is based on traditional IR models, such as, the vector space model, BM25, etc.

Contextualization. This model uses information from an element itself, but also from its context (ancestor or descendant elements). This new characteristic results in a better retrieval performance, in comparison with the *element scoring model*, which only considers the element itself.

Propagation. When a *leaf-only* indexing strategy has been used there is only statistics for leaf elements. Consequently, a propagation mechanism is needed to calculate the relevance score of non-leaf elements. This propagation combines the scores of the leaf elements (usually by a weighted sum) and any additional element information (such as its position, distance, etc., within the XML hierarchy).

There is still two important tasks when dealing with structured IR:

Processing structural constraints. There are two different ways to process structural constraints: a *strict* or a *relaxed* approach. Within the *strict* approach, the structured IRS is not allowed to return elements not exactly matching those specified in the query. However, within the *relaxed* approach, other elements different from those specified in the query may be retrieved. The main components used to comply with a relaxed approach are the construction of a dictionary of tag synonyms and to perform a structure boosting in the retrieval process. The dictionary of synonyms allows to retrieve semantically related elements, such as <*home*> or <*house*>, independently of which one has been specified in the query. Within a structure boosting approach, the retrieval score of an element is computed ignoring the structural constraints, but boosted if the element matches those structural constraints.

Processing overlaps. An overlap occurs when an element text is contained within another element. A structured IRS usually returns a non-overlapping list of results to ensure the user does not receive the same text under different output elements. When overlaps happen a decision must be taken on which of the overlapping elements is the best answer. This decision usually depends on the application and/or the user preferences. Removing overlap is usually done after the ranking process, and it may be done in different ways. The most simple and used approach is to keep the highest ranked overlapped element and remove the others. A more sophisticated approach, shown to outperform the previous one, is to analyse the distribution of retrieved elements for each document, to decide which ones to return, e.g., if all sections of a chapter and the chapter itself has been retrieved, it would be better to return only the chapter although any of the sections would have a higher ranking than the chapter.

3.5 Presenting results

Any of the previous section retrieval models provides a ranked list of retrieved results, sorted decreasingly by the model assigned relevance values (unless the boolean model), as the answer for the user issued query. Non-structured IRSs simply show this retrieved list as it is to the user in the IRS interface. For structured IRSs this task is a bit more difficult, since the retrieved XML elements are not independent,

as flat documents in traditional IR. There are four different tasks to show the XML retrieved elements, also used in INEX tracks:

Thorough task. It presents the results exactly as they are retrieved from the IRS. Therefore, some overlaps may occur providing to the user redundant information.

Focused task. It presents the most focused document elements to the user without any overlap.

Context tasks. These tasks offer the user the *focused task* retrieved elements grouped by document, in their original document order. These tasks are intended to facilitate the user navigational access to the retrieved elements, assuming the user prefers documents as the retrieval units, thus giving an overview of relevance in context. There are two different context task approaches: 1) *Relevant in context* returns non-overlapping XML relevant elements grouped by the document to which they belong, and 2) *Best in context* returns a single document XML element, as the best entry point for starting to read the relevant content in the document.

3.6 Information retrieval evaluation

An important concept for IR evaluation is *relevance*. A document from the document collection is considered as *relevant* or *not relevant*, usually a binary classification, depending on whether or not this document solves the user information need (expressed by the user query). All these document relevancy decisions are known as *relevance assessments*. Since these relevance assessments are usually manually done by experts, they are considered as the gold standard or ground truth judgement of relevance.

IR evaluation consists of measure the retrieval performance of an IR system or model. The basic followed process to evaluate the IRS retrieval effectiveness is as follows: for a given query, the retrieval model (to be evaluated) returns a ranking of results ordered by a system decreasing relevance degree. A similarity measure is then calculated, comparing these results against the relevance assessments for this given query. The higher the similarity, the better is the retrieval effectiveness of the evaluated model.

An evaluation methodology based on the Cranfield paradigm, proposed by Cleverdon et al. [28], is generally used to evaluate system-centred IRSs. This evaluation framework consists of a *test collection* composed by a document collection, a set of well-defined queries, and a set of manually assigned relevance assessments for each of those queries. The use of this framework under different IR evaluations, but using the same test collection, allows to obtain repeatable and comparable results, together with generalizable conclusions, from the evaluation process. The previous facts are very important, in order to be able to compare different IRSs or to keep improving a given one.

There are different test collections, such as: the Cranfield collection, which was the first serious approach, but it is too small nowadays. The TREC (Text REtrieval Conference) collection[8], being used since 1992 in the previous conference. It is composed by 1.89 million documents and relevance assessments for 450 information needs. This is currently the state-of-the-art test collection in IR. And other test collections as GOV2[9], a very large Web page test collection, or NTCIR[10] and CLEF[11] more focused on cross-language IR.

In order to measure the effectiveness of any system, some *evaluation metrics* must be used. The most common evaluation metrics are *precision* and *recall*. Next, their formal definitions are presented:

Precision (P) is the proportion of retrieved relevant results (res_r) from the total number of retrieved results (res), being expressed as:

$$P = \frac{res_r}{res}, \tag{3.4}$$

and, *Recall (R)* is the proportion of retrieved relevant results (res_r) from the total number of relevant items in the collection, (tot_r), being expressed as:

$$R = \frac{res_r}{tot_r}. \tag{3.5}$$

The higher both metrics are, the better is the IRS retrieval model. But both metrics tend to be inversely proportional, so a trade-off between them is needed.

[8] http://trec.nist.gov/data/test_coll.html
[9] http://ir.dcs.gla.ac.uk/test_collections/
[10] http://research.nii.ac.jp/ntcir/data/data-en.html
[11] http://www.clef-initiative.eu/

However, in some situations one is preferred to the other. For example, in web search, precision is more important (since the corpus is extremely huge), but in some expert domains (e.g. legal/medical environments) with smaller corpus, recall is more important (since anything relevant need to be known). Precision usually decreases and recall increases as the number of retrieved documents increases.

F-measure is the weighted harmonic mean of precision and recall. It is used as a trade-off precision versus recall metric, and it is expressed as:

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}, \tag{3.6}$$

where values of $\beta < 1$ emphasize precision, values of $\beta > 1$ emphasize recall, and with $\beta = 1$ both precision and recall have the same importance, being also known as F_1 in this last case. For F_1 the right part of the equation is equal to $2PR/P + R$.

The previous metrics are suitable for unranked retrieval results, such as those provided by the boolean retrieval model. However, other metrics are needed to measure ranked list of results, where usually only a set of the top k possible results are returned, and their order is important. Next, we present some of these metrics.

Average precision (AP) is an approximation to the precision-recall curve (*p(r)*) for a given ranked result list, which computes the average value of *p(r)* over the interval from $r = 0$ to $r = 1$. AP can be expressed as:

$$AP = \frac{\sum_{k=1}^{n} P(k) \cdot rel(k)}{\#relDocs}, \tag{3.7}$$

where k is the position of the element within the rank, n is the number of retrieved documents, $P(k)$ is the precision at cut-off k in the list, and $rel(k)$ is equal to one if the element at rank k is relevant and zero, otherwise. And finally, $\#relDocs$ is the total number of relevant documents in the document collection. This measure may be interpolated to reduce the sawtooth shape in the curve.

Mean Average Precision (MAP) is simply the AP average over a set of queries Q, being expressed as:

$$MAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q}. \tag{3.8}$$

AP and *MAP* provide precision measures at all recall levels, but sometimes this is not interesting, such as in web retrieval, where the interest is to have many relevant results at first positions.

Precision at k (P@k) is the precision at position k of the retrieved list. It is a very easy to calculate measure, but it is also the more unstable of the commonly used metrics. This is because it highly depends on the total number of relevant documents for a given query, which considerably varies between queries. For that reason, its averaged value between queries is not a very good effectiveness estimation.

R-precision alleviates the previous problem, since it requires to know the set of relevant results *Rel*. R may be expressed as:

$$R = \frac{r}{Rel}, \tag{3.9}$$

where r is the number of relevant results, among the first *Rel* results in the retrieved list.

Finally, a lately widely used metric is the *Normalized Discounted Cumulative Gain (NDCG)* [62]. It 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 is normalized by the ideal ranking, where all the relevant results would be consecutive starting from the first position. With this normalization, the metric values are always between 0 and 1, making possible to calculate averages among different queries. The metric value for a given list of results is calculated as follows:

$$NDCG@x = \frac{1}{N} \sum_{i=1}^{x} \frac{2^{rel(d_i)} - 1}{\log(i+1)}, \tag{3.10}$$

where x is the evaluation threshold, N is the ideal DCG for the relevant results (all relevant results consecutive from the list first position), i is the ranking position of the result being evaluated, d_i is the result at position i, and $rel(d_i)$ is the relevance value of d_i. $rel(d_i)$ would be 0 if the document has been judged as *not relevant*, and 1 if it has been judged as *relevant*, or any preassigned values for non-binary relevance assessments.

All the previous evaluation metrics are designed for non-structured documents, but they can be adapted to work with ***structured documents*** too. As in the case of traditional IR, there are also some test collections for structured IR, such as, the *Shakespeare* collection[12], which was the first used approach, and the *INEX* collection, which is the state-of-the-art test collection for structured IR nowadays.

The main characteristics of an XML test collection are the following: the documents are obviously in XML format, there are CO and CAS queries, and the relevance assessments are performed at the XML element level, also measuring if the system retrieves the right document structural units. The document collection on the last editions of INEX is composed by the English Wikipedia articles in XML. The main retrieval tasks are the focused, relevant in context and best in context tasks, already explained in Section 3.5.

Within INEX, the evaluation methodology is based on different aspects of focused retrieval. The relevance assessments and evaluation measures only consider the amount of highlighted text in relevant documents, which allow to use metrics that are natural extensions of the previously seen traditional metrics. As the explanation of all the adapted metrics used in INEX would be very long and repetitive, considering the above traditional IR metrics where they come from, they can be checked in [64] for a deeper information.

3.7 The GARNATA IRS for XML retrieval

Garnata is a structured IRS based on probabilistic graphical models, concretely on an *Influence Diagram (ID)* [88], a generalization of the well-known *Bayesian Network (BN)* formalism [96], in the context of Decision Theory [47]. It has been designed and developed by our research group, being registered in the Intellectual Property Registry of Andalusia in 2012. It is written in C++ following the object-oriented paradigm, offering a wide range of classes and a complete set of utility programs. It has been designed with especial interest on efficiency, using an efficient combination of data structures that ensure a good time response for a query.

Garnata is designed to work with structured information, concretely with XML documents. This structured IRS has been improved and tested at three editions of

[12] http://xml.coverpages.org/bosakShakespeare200.html

the INEX Workshop ([38] describes the last participation). It has also been applied to build a real IRS for parliamentary documents[13], being a previous version of this IRS explained in [40].

In contrast to traditional IR, the retrieval of a document component in structured IR is not independent of the retrieval of other components. Other component factors must be taken into account in the retrieval process, such as, their usefulness for the user, their context within the document structure, or what else has been previously retrieved.

Garnata implements the *Context-based Influence Diagram (CID)* retrieval model for structured documents, described by de Campos et al. in [34, 35]. It provides a visual representation for a structured retrieval task, presented as a decision-making problem. To solve an ID means to determine the expected utility of each one of the possible decisions, for those situations of interest, with the aim of making decisions which maximize the expected utility, as Shachter describes in [111]. CID is able to make decisions about the best document components to be retrieved, considering not only their *probabilities of relevance*, but also their *utilities* for the user (user preferences) and their context, such as, their location within the document structure.

The final relevance value assigned by CID to a document component is computed based on two different types of information. On the one hand, the *specificity* of the component with respect to the query: the more terms in the component which appear in the query, the more relevant the component becomes. That is to say, the more clearly the component is only about (at least a part of) the topic of the query. On the other hand, the *exhaustivity* of the component with respect to the query: the more terms in the query which match with terms in the component, the more relevant the component is, i.e., the more clearly the component comprises the topic of the query. The components which best satisfy the user information needs expressed by means of the query should be, simultaneously, as specific and exhaustive as possible.

In this chapter we have presented a formal overview of all the main stages of an IRS. We have gone through the different IR process stages, firstly explaining them for plain documents as the information basis, and then showing how to extend them to deal with structured documents, while highlighting this new approach

[13] http://irutai2.ugr.es/SEDA/

difficulties. We have started explaining how documents content is represented, pre-processed and indexed in an efficient way. Once the index has been built, we show how users can send content-only (CO) or content-and-structure (CAS) queries to the system, to try to solve their information needs. Then, we have explained how different retrieval models match these queries against the built index to return a ranked list of retrieved results. In the following section, we have explained different ways to show these results to users, depending on their visualization preferences. The next section is dedicated to explain how these IRSs are evaluated, including several metrics to measure their retrieval effectiveness. And finally, the basics of Garnata, the structured IRS most of the times used in this thesis, are explained in the last section.

Part III

Research Contributions

Chapter 4

Personalization Techniques for XML IR

4.1 Introduction

As the amount of information increases exponentially every day and 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 interests and preferences. On the other hand, the information is being stored more and more in a semi-structured way, and XML has emerged as the standard for representing and exchanging this type of data. XML search allows a higher retrieval effectiveness, due to its ability to retrieve and show the user the specific relevant parts of documents, instead of the whole document. The joint use of personalization techniques along with XML IR offers users a saving in both time and effort, in the search process to solve their information needs, since personalized XML IRSs provide specific relevant results adapted to each user interests and preferences.

This chapter is devoted to the development and evaluation of new personalization techniques in the context of XML retrieval, 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 the search (query reformulation, in our case, query expansion and transformation on content-and-structure queries), after the search (reranking of results) and within the retrieval

process (modification of the retrieval model).

The developed personalization techniques are mainly designed for structured document collections, such as digital libraries or corpus of big organizations, more than for the web due to its high structural heterogeneity. However, most of the proposed personalization techniques could also be applied to flat (non structured) documents with almost no changes and effort. With respect to the user profiles, in this chapter we focus on their stored information effective use, i.e. how good the whole retrieval process is in order to exploit the information stored in the user profile, rather than on their construction process (see Chapter 6 for this purpose).

The main contribution of this chapter is the proposal and evaluation of several new personalization techniques designed for 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 of the search engine, or even the use of 'content and structure' queries for personalization purposes. Observing the obtained experimental results, based on a user study using a parliamentary document collection, we can conclude that all of them provide very good performance improvements over not using personalization. We suggest to use the proposed techniques, if possible, in this order: the retrieval model modification, the content and structure approach and the reranking approach.

The rest of the chapter is organized as follows. We first give, in Section 4.2, an overview of the different structured personalization strategies existing in the literature. In section 4.3, we show our proposed personalization approaches. Section 4.4 shows the used XML document collection and the carried out user study to get the relevance assessments. Section 4.5 describes the experimental environment, including our evaluation methodology and the obtained results and conclusions. Finally, we finish in Section 4.6 with some general conclusions and proposals for future work.

4.2 Related work

This section and chapter focus on XML personalization techniques, since this is the field in which this thesis is framed. Section 2.3 should be checked for an overview

of general personalization techniques. We next comment some aspects of querying XML documents, and then describe specific XML personalization methods.

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 INEX (INitiative for the Evaluation of XML retrieval), keyword queries are known as content-only (CO) queries, whereas content-and-structure (CAS) queries are those containing both structure and content constraints (see Section 3.3 for more information).

Although there are other querying languages for XML IR, we use NEXI (Narrowed Extended XPath I) [130] in this thesis, which allows us to retrieve XML documents based on content and structure. The drawbacks of these languages are that they are complex to learn to use, and users must know the structure of documents, which most of the times is not the case. These query languages are more suitable for expert users, letting them to better specify their information needs in comparison with the classic keyword search.

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. [4] 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[1], and the 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 active participation.

Chernishev [21] takes PIMENT architecture as the base, adding a feedback module which tries to extract, from query history, the user 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

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

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 process, it uses a mechanism based on a modified and well-known technique of query rewriting called relaxations.

Amer-Yahia et al. [5] 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, also defining an OR-aware top-k pruning algorithm to guarantee an efficient query personalization process.

Our approach for XML personalization is fairly different from these previous 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. One of our proposals for personalization is also based on transforming the original CO query into a CAS query that incorporates the profile terms.

Relevance feedback and blind relevance feedback techniques, although different, are related in several ways to personalization. Therefore, it is also interesting to briefly review existing work on these techniques over XML documents. Within this area, Mass and Mandelbrod [77] 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 [93] 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. [36, 39] 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. [57] 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 [109, 110] 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 Hlaouna et al. [56].

4.3 Developed personalization techniques

In this section we are going to describe the different developed approaches 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 into 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. There are strategies for each of the three typical scenarios where personalization can be implemented: before, within and after the search is performed.

These approaches will be experimentally compared in Section 4.5. One of the principles guiding our research is that we want most of the work to be carried out by the IRS search engine, i.e., we try to avoid the use of expensive additional processes or calculations in order to integrate the user profile information (as in many of the personalization strategies mentioned in Sections 4.2 and 2.3).

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 scores assigned by the system. Each IRS result is an *structural unit* of an XML document in the collection. The list of results contains, at most, a fixed number of SUs (e.g. 1500 in the Section 4.5 experiments) and follows the "Focused" INEX Task specification [64], i.e., overlapping has been removed.

4.3.1 Normalized query expansion (NQE)

The first approach we are going to use is simply a 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 higher importance. The number k of added terms is a parameter that should be adjusted.

This is a very easy and efficient technique, which only requires to perform a longer query. But its main drawback is the *query-drift* problem, by which the retrieved results may not contain the query terms the user was looking for in the original query (see Section 2.3.1). The expanded (*original+profile*) query could retrieve results closer to the user profile itself than to the original query (which represents the user actual information needs). 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 actual 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 than the original query terms. 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 ($p_i = 1,\ i = 1, \ldots, m$) by default, and the expanded profile terms with the following weights,

$$p_i = p_0 * \frac{w_i}{\max_{m+1 \leq i \leq m+k} w_i},\ i = m+1, \ldots, m+k. \tag{4.1}$$

Table 4.1 shows an example of the NQE application. As shown in this table, the user profile w_i term weights may have any values, from very high to very small, which will depend on the followed method to calculate them. It is obvious that the simple aggregation of these weighted profile terms to the original query terms will

Table 4.1: Two examples of the expanded final query using and not using *NQE*, where the original query terms are '*olive oil*', and NQE is applied with $k = 3$ and $p_0 = 0.66$ over the very low and very high user profile term weights, respectively.

No NQE	1.0*olive 1.0*oil 0.006714*agriculture 0.006580*farmer 0.004048*production
NQE	1.0*olive 1.0*oil 0.66*agriculture 0.647*farmer 0.398*production
No NQE	1.0*olive 1.0*oil 2.066*agriculture 1.822*farmer 1.535*production
NQE	1.0*olive 1.0*oil 0.66*agriculture 0.582*farmer 0.49*production

not produce the desired results. This is because the influence in the retrieval process of the expanded profile terms, with respect to the original query terms, could also be too large or too low, almost deleting the influence of the original query terms or almost not influencing at all, respectively, in the expanded query retrieved results. This is another perspective of the aforementioned *query-drift* problem. NQE tries to avoid this problem normalizing the user profile term weights, with respect to the original query term weights. Therefore, the added profile terms can receive, at most, a fraction p_0 of the default weight attached to the original query terms. The normalization factor p_0 is another parameter to be adjusted.

4.3.2 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 it 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 [79].

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 terms. Thus, a higher 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 some kind of *context*.

However, what we could do is to replace in the previous approach the profile query with the expanded query obtained by *NQE*, 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 higher 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, also helping 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[2]. The basic idea is to reward SUs in the original query results that match with any SU in the expanded query results.

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 a SU in a list overlaps with a different SU in the other list. In XML retrieval, if a SU is relevant, then its container or descendant SUs are also relevant (at least relevant to 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 (Figure 4.1 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

[2]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 [79].

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 Figure 4.1, 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 highest 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 being 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.

4.3.3 Structural query expansion: CAS queries

Another approach for personalization would be to perform a sort of query expansion, but exploiting the structural characteristics of XML to build the expanded query.

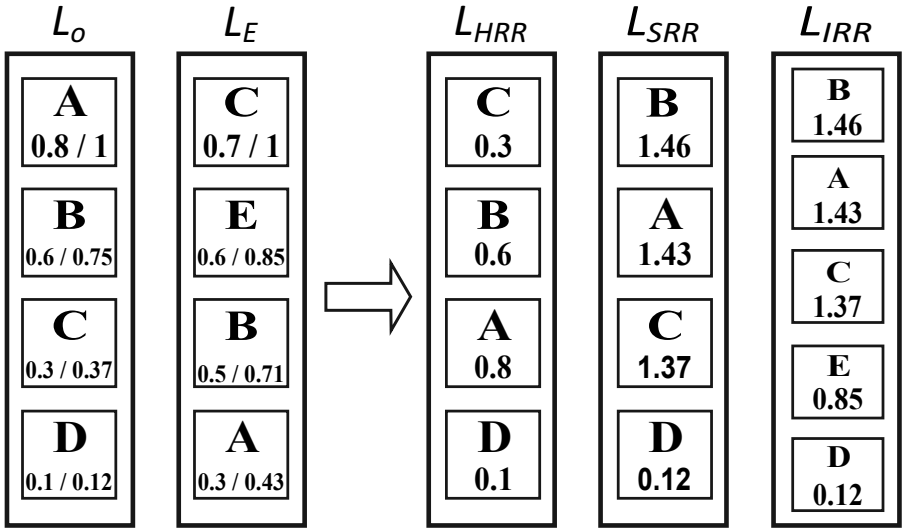

Figure 4.1: Example of how the proposed reranking strategies work. The numbers associated with each SU correspond to its *original/normalized* RSV values.

CAS queries allow us to exploit 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.

In order to allow CAS queries to be specified, we have selected the NEXI language [130], widely used within INEX. The general form of a NEXI CAS query is *//A[B]//C[D]*, which "returns C descendants of A, where A fulfills the condition B and C fulfills the condition D" (see Section 3.3 for more information about CAS queries).

We are going to transform the original CO query into a CAS query, in such a way that its target part (*//C[D]*) coincides with the original query, and its context part (*//A[B]*) 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 any descendant from root), 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 SU in the collection, `MaxUnit` (which is the less restrictive SU to hold the profile terms). Therefore, the expanded CAS query would be as follows,

```
//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(.,profileTermK)]//*[about(.,originalQueryTerms)]
```

4.3.4 Modification of the retrieval model

All the previous personalization strategies 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 a very common practice in personalization strategies (specially in web personalization, where the search engine cannot be modified, mainly because it is usually inaccessible to the researchers).

This strategy depends completely on the retrieval model underlying the search engine being considered. In this case, we have used Garnata (see Section 3.7 for further information). Nevertheless, it is possible that the ideas underlying this modification of Garnata can be applied to other IRSs.

To understand how we have modified the Garnata search engine underlying retrieval model, it is necessary to briefly remind how it computes the RSV values of each SU in a document (check again Section 3.7). It combines two different types of information, the *specificity* and *exhaustivity* of the SU with respect to 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 a 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 terms in the query that also 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,

$$util_{Q,n}(U) = nidf_Q(U)\frac{e^{\left(nidf_Q(U)\right)^n} - 1}{e - 1}, \tag{4.2}$$

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, and n is a parameter that controls (in a non-linear way) the extent to which more terms from the query must be contained in a SU, in order to get a high utility value for this unit. In this way, the higher 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 high 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 a SU having only 1 original term and 15 profile terms, instead of a 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 NQE 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.

4.4 Common experimental components

In this section we are going to explain the core common experimental components used in all chapters of this thesis Part III (*Research Contributions*). This common experimental components are the XML document collection and the carried out user study (including queries, users and user profiles) in order to obtain the relevance assessments.

4.4.1 XML document collection

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 this thesis we have used a document collection composed by documents in XML format from the Andalusian Parliament. The AP was established in 1982 and until now, there have been nine legislatures (political periods of up to four years). Our research group has been collaborating with the AP Official Publications Service[3] since 2005. Along this collaboration they have been providing us with their two main official publications: the *records of parliamentary proceedings (sessions)* and the *official bulletins*. The first publication contains the full transcriptions of the Members of the Parliament (MP) speeches in each parliamentary session, where laws are passed or different issues of interest are discussed. The official bulletins publish all the information likely to be public, such as passed laws or any other interesting information.

[3] http://www.parlamentodeandalucia.es/opencms/export/portal-web-parlamento/composicionyfuncionamiento/serviciosadministrativos/publicacionesoficiales.htm

There are three different types of sessions in the AP: *plenary sessions* attended by all MPs to discuss an initiative, *committee sessions* attended by MPs belonging to different areas of interest (agriculture, education, employment, etc.) to discuss a relevant initiative, and *permanent parliamentary sessions* attended by some duty MPs when AP is not in ordinary session. AP works around the initiative concept, where a proposal of a MP or a political party is discussed in a session.

Since our objective is to use the document collection under personalization purposes, we saw convenient to focus on the *committee sessions*, which are devoted to different areas of interest, which at the same time could represent different user interests or profiles. In 2011, when we started this thesis preliminary research works, the number of committee sessions we had ready to be used was 658 xml documents. All the previous documents belong to the sixth and seventh legislatures (from March 2000 to March 2008), containing a total of 432575 different SUs and having a size of 122MB.

Next, we are going to describe the internal structure of the AP records of parliamentary proceedings, where committee sessions belongs. They have two different and well defined parts: the first part is a general information section containing for example the legislature number, type of session, date, or president. And the second part is the development of the session, with the grouped by type list of initiatives, each one with fields such as, the initiative type, proposer(s), results of any vote, MPs participating in the debate and their respective speech transcriptions, etc. All this information is specified in the schema (concretely DTD) of this type of documents, which we show in Figure 4.2.

4.4.2 User study

As we need relevance assessments in order to evaluate the performance of the different personalization strategies, we have carried out a user study. The goal of this user study is to obtain the relevance assessments for each combination of query, user committee-based selected profile and user, which we denote as an *evaluation triplet*. We have followed the advices given in [12, 24] for ensuring the reliability of the user study outputs. We next explain all the different components of this carried out user study.

```
<!ELEMENT diario_sesion_pa (legislatura, numero_diario, tipo_sesion?,
  organo, presidente, numero_sesion?, fecha, desarrollo)>
<!ELEMENT legislatura (#PCDATA)>
<!ELEMENT numero_diario (#PCDATA)>
<!ELEMENT tipo_sesion (#PCDATA)>
<!ELEMENT organo (#PCDATA)>
<!ELEMENT presidente (#PCDATA)>
<!ELEMENT numero_sesion (#PCDATA)>
<!ELEMENT fecha (dia, mes, anio)>
<!ELEMENT desarrollo ((epigrafe | iniciativa)+)>
<!ELEMENT dia (#PCDATA)>
<!ELEMENT mes (#PCDATA)>
<!ELEMENT anio (#PCDATA)>
<!ELEMENT epigrafe (tipo_epigrafe, iniciativa+)>
<!ELEMENT tipo_epigrafe (#PCDATA)>
<!ELEMENT iniciativa (tipo_iniciativa, numero_expediente?, extracto?,
  proponentes?, debate_agrupado?, tramite?, votacion*, intervienen?,
  materias?, intervencion+)>
<!ELEMENT tipo_iniciativa (#PCDATA)>
<!ELEMENT numero_expediente (#PCDATA)>
<!ELEMENT extracto (#PCDATA)>
<!ELEMENT proponentes (#PCDATA)>
<!ELEMENT debate_agrupado ((componente)+)>
<!ELEMENT componente (numero_expediente, extracto?, proponentes?)>
<!ELEMENT tramite (#PCDATA)>
<!ELEMENT votacion (#PCDATA)>
<!ELEMENT intervienen (#PCDATA)>
<!ELEMENT materias (#PCDATA)>
<!ELEMENT intervencion (interviniente,discurso)>
<!ELEMENT interviniente (#PCDATA)>
<!ELEMENT discurso (parrafo+)>
<!ELEMENT parrafo (#PCDATA)>
```

Figure 4.2: DTD specification for the AP records of parliamentary proceedings, where committee sessions belong.

Table 4.2: The user study 23 queries (translated into English).

musical activity	central america residues management
seville olive cultivation	coast landscape degradation
water purification	disability employment
public employment	disease virus transmission
andalusian exports	economic expenditure scholarships
andalusian gastronomy	computer science
granada province investments	stem cell research biotechnology
personal income tax	loja
ejido west almeria	andalusian product promotion
security and new technologies	prices rise
breast cancer preventive treatment	internet web use administration
scheduled visits	–

Queries. We have used an heterogeneous set of 23 queries, shown in Table 4.2. These queries have been formulated by real users of the Section 4.4.1 document collection. Hence, they represent a small-medium but trustworthy sample of real user information needs. This set of queries has an average length of 2.61 terms per query (in the Spanish original version), which is in the range of the average search query length studies. For example, Jansen et al. [61] in 2000 shows an average search query length of 2.4 terms, but recent studies, such as Taghavi et al. [124] in 2012 has found that this value has grown over time showing a value of 3.08 terms per query.

User profiles. Since our main focus is on the behaviour and retrieval performance of the different personalization techniques, it is not the goal to build the best possible user profile. Therefore, a simple approximation to build the user profiles has been initially considered. We use weighted keyword-based as the representation of our user profiles (see Section 2.2.2). These weighted keywords may be automatically extracted from documents, other kind of sources or directly provided by the user. We have concretely learned them from the document collection documents. This is possible because these corpus documents are classified into different areas of interest. We have learned a profile for the eight most represented document collection areas of interest-committees: agriculture, culture, economy, education, employment, environment, health and justice.

Table 4.3: The first ten terms and *idf* weights corresponding to the eight selected user profiles. The terms are translated into English and unstemmed.

Agriculture	1.822*agriculture 1.535*sector 2.066*agrarian 2.068*fishing 1.965*production 1.659*help 2.220*farmer 1.839*product 2.351*oil 2.098*rural
Culture	1.760*culture 2.091*sport 2.015*heritage 2.041*tourism 1.951*cultural 2.385*museum 2.183*tourist 1.988*history 1.189*knowledge 1.165*andalusian
Economy	1.367*economy 1.421*budget 1.900*account 1.338*freelance 1.487*million 1.633*euro 2.136*treasury 1.889*year 1.810*finance 1.790*fund
Education	1.457*education 1.368*centre 1.968*student 2.063*professor 2.114*teach 1.069*council 2.071*course 2.252*school 2.556*conservatory 1.763*training
Employment	1.487*employment 1.244*job 1.818*labour 1.050*social 1.542*company 2.154*prevention 1.993*contract 1.763*training 2.106*risk 1.411*service
Environment	1.845*environment 1.351*milieu 1.869*natural 2.103*park 2.556*forested 1.274*plan 2.384*environmental 0.831*mass 1.669*zone 2.565*fire
Health	2.108*hospital 1.719*health 1.992*sanitary 1.411*service 1.368*centre 2.390*doctor 1.717*care 1.185*public 2.435*patient 1.001*group
Justice	1.163*law 1.217*government 1.069*council 1.982*justice 1.001*group 1.338*autonomous 1.739*local 1.388*manage 1.500*council 1.237*policy

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. Table 4.3 shows the first ten terms for the eight selected user profiles.

The process to learn these user profiles based on the different areas of interest document collection content is quite similar to the indexation process, so it would be a good idea to include this user profile learning process within the indexing process. In this way, each time some new documents are included in the index (index update), the corresponding user profiles will be also updated with these new documents content information. If this is not the case, the process of learning user profiles must be carried out every time the index is updated.

The reader may think the user profiles are not very 'real', because any user would be interested in some different areas of interest. For simplicity, we have labelled the user profiles with only one word (or category), but some of them are actually about different related issues. For example, the *agriculture* profile actually contains information about agriculture, livestock and fishing, and the *culture* profile contains information about culture, tourism and sports. Any other user profile

areas of interest configuration may be perfectly valid and usable with the developed personalization techniques, even real user profiles if they were available.

The user study. In this user study we have kept a simple keyword web search query interface, although the IRS (Garnata) exploits XML structure during the query processing, so that the retrieved results can be any kind of document SUs. We have decided to take this approach, because some users from the user study did not know the structure of the underlying XML document collection, and most of them did not know any of the complex structured querying languages. The resulting relevance assessments were made after a brief training phase to familiarize the users with the IRS interface.

The user study involved 31 users. Each user selected the Table 4.3 profiles he/she was more comfortable with, corresponding to a person interested in documents related to the topics discussed in these specific committees. This choice was taken with the only information of a brief explanation of the main topics discussed in each committee, but not with the user profile learned terms as shown in the previous table. It was done this way to do not bias the user relevance judgements when these terms appear in the evaluated results. Each user submitted and evaluated one or several of the previous 23 queries to the IRS, but always assuming only one of the selected profiles for each query. When a user evaluated a query under a given profile, a set of relevance assessments was obtained for this user, profile and query[4]. Henceforth, we will refer to this file of relevance assessments as an *evaluation triplet*. A total number of 126 different *evaluation triplets* were obtained from the user study.

Following the guidelines of [91], and considering the number of triplets, we ask the users to judge deep pools (under the corresponding profile) for the selected topics, ensuring that if the user starts judging a query, he/she completes the full assessment process for it. Particularly, a pool of up to 100 elements has been considered, pool size that has been proved to give reliable results [143].

The pool is composed by the 50 first elements retrieved by the non personalized IRS in response to the query, plus the 50 first results returned by the IRS using the *HRR* personalization strategy for the same query. We did it in this way to avoid that

[4]It should be noticed that the user did not judge if a given retrieved result was the best possible one, but only whether or not its content was relevant to the given query and profile (binary assessments).

many possible relevant results, not appearing among the first 50 results obtained by the non personalized query, were considered as irrelevant[5]. The previous judgements were performed separately, randomly between the original IRS and the personalized IRS approaches. After that, we had two lists of relevant results. In order to have a unique list for each evaluation triplet, we fused these two lists, deleting duplicates and overlaps (maintaining the larger 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 personalization strategy used to perform the second part of the evaluation was selected carefully, to avoid the bias that the relevance judgements obtained with this strategy could induce on the evaluation of the other strategies. As *HRR* only reorders the list of results of the original non personalized query, it does not introduce any relevance assessments not present in the original results list.

As a broad overview of the resulting evaluation triplets, the average number of relevant results in the pool per query is 18.7, with a standard deviation of 14.2. Note that this high value for the standard deviation mainly comes from the fact that we are considering different profiles for judging the same query. For instance, the query "prices rise" has an average number of relevant results of 30, 2, 45 and 2 under the profiles of agriculture, culture, economy and education, respectively.

4.5 Experimental evaluation and results

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

[5]The source of the problem is the limitation of judging only the first 50 results retrieved by the IRS, but it was necessary since the evaluation of a higher number of results would require too much time and effort from users.

strategies will help the user to find the previously judged relevant components more easily, by comparing the results obtained with and without them.

4.5.1 Evaluation metrics and structural adaptations

For the previous evaluation purpose, we have considered particularly valuable the use of rank-based measures. Concretely, the *NDCG* as the evaluation metric, which better fits our requirements. A detailed explanation of this metric can be checked in Section 3.6. As a reminder, NDCG is designed for estimating the cumulative relevance gain a user gets examining the first documents in a retrieved list of results. It has a discounting factor as the user examines more results in the ordered list of results. The metric formula is:

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

where x is the evaluation threshold, being $x = 50$ in our experiments. With the normalization, the metric values are always between 0 and 1, making 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, we have proposed the following two NDCG metric considerations/adaptations, which 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 irrelevant, that is, $rel(d_i) = 0$. However, what does one do when a SU d_i overlaps with some of the SUs judged as relevant by the user (what we call 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 a SU which matches any relevant SU from the corresponding evaluation triplet is also relevant (to some degree).

Table 4.4: 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

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 our work we follow the second approach. Let us explain the reasons for this choice. In our document collection there are four retrievable SUs (may be checked in Figure 4.2): *proceedings (desarrollo)* (the complete document corresponding to one session of a Committee), *initiative (iniciativa)* (the debate of a parliamentary motion within a session), *intervention (intervencion)* (of a member of the parliament in the debate of a motion) and *paragraph (parrafo)* (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 a 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 4.4.

- **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 the Equation 3.10, calculated as the ideal DCG value for the relevance assessments, is

$$N = \sum_{i=1}^{\min(x,rj)} \frac{1}{\log(i+1)}, \tag{4.3}$$

where rj is the number of results judged as relevant by the user for this evaluation 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) a 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 (N) and the number of SUs considered in the numerator of Equation 3.10 is not coherent. In the first case, only one retrieved SU contributes to the summation of Equation 3.10, 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 larger 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 a SU, larger 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 larger SU. In this case we add $v-1$ units to rj.

An example of the NDCG structural normalization calculation can be seen in Figure 4.3. As we can see, there are $rj = 4$ relevance assessments in this 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 to rj). Finally, the third, fourth and fifth results match the same relevance assessment (so we add 2 to rj). 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,

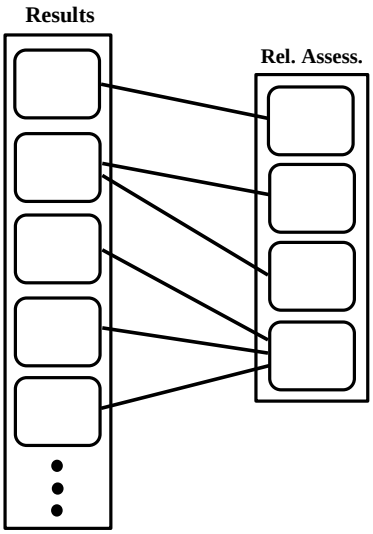

Figure 4.3: Example of NDCG structural normalization process.

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)* [104], also called robustness index in [16], which in our context is defined as: the ratio of the difference between the number of evaluation triplets helped (n_+) and of those hurt (n_-) by the personalization strategy, to the total number of evaluation triplets, $nt = 126$, expressed as follows:

$$RI = \frac{n_+ - n_-}{nt}. \tag{4.4}$$

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

4.5.2 Results

This section shows all the obtained results from the performed experimental evaluation. Additionally to the Section 4.4 common experimental components and the previous Section 4.5.1 evaluation metrics, concretely in this chapter, we have also considered a different way of building the user profiles. In this case, we have asked

some expert users of the document collection to manually build the profiles, by selecting and ordering the terms they think better represent each of the considered areas of interest (although the weights are still based on *idf*). The idea behind the use of these *expert profiles* is to see to what extent the automatically build user profiles are a good representation of the different document collection areas of interest, and to see if the different personalization techniques results by using both kinds of user profiles considerably diverge or not.

Tables 4.5 and 4.6 show the NDCG and RI values obtained from the different experiments when using the automatically generated profiles under the Garnata IRS. These tables NDCG results can also be graphically observed, for most of the personalization techniques, in Figures 4.4 and 4.5. In the last figure, SRR subfigure also represents IRR, since their NDCG values differences are very small and they would not be well appreciated. Additionally, in the CAS/CAS-or subfigure, the y axis is rescaled in order to be able to see these personalization techniques very small performance differences. 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 techniques considered in Table 4.5 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 ($I\text{-}HRR$). 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 $p\text{-}HRR$), 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 only the *profile* query, by comparing the performance of HRR and p-HRR.

The personalization techniques considered in Table 4.6 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*).

In total, we have performed the evaluation with a comprehensive set of 13 different personalization techniques. Being a comprehensive set because within this group, there are strategies which act in the three possible places any personalization technique may be performed in the retrieval process: before, within or after the search is executed, even representing some of them hybridization approaches over the previous three retrieval phases.

In all the cases, we have experimented with four different values of k, the size of the set of expansion terms (5, 10, 20 and 40) – only the first three values in the case of expert profiles, since most of these profiles do not have enough terms to use $k = 40$. 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 techniques (except with QE, which does not use it).

Considering all the previous parameters, we are able to calculate the total number of issued queries to the IRS, and therefore all the evaluated retrieved lists of results in these experiments (for automatic user profiles). Each of the 126 evaluation triplets must be performed for each proposed personalization technique (13), which may be fed with 12 different user profile configuration parameters $k * p_0$ (unless QE which does not use p_0), plus an additional run for the original queries, i.e., $126 * ((13 * 12) - 8(QE) + 1(Orig)) = 18774$ retrieved result lists to be evaluated. Each of these lists provides NDCG and RI values, after being evaluated using the corresponding evaluation triplet set of relevance assessments. But, as 18774 NDCG-RI values are totally unwieldy, for each personalization technique and user profile configuration values (k and p_0), the average (for NDCG) and the result from Equation 4.4 (for RI) among the 126 evaluation triplets are calculated, being the values shown in the Tables 4.5 and 4.6.

A global average (μ) and standard deviation (σ) is also calculated for the previous NDCG and RI values corresponding to the different user profile configuration parameters for each personalization technique. The best NDCG and RI values for each personalization technique appear in *bold*. For the NDCG values, if *both* a *paired*

Table 4.5: NDCG and RI values obtained in the experiments with QE, NQE, HRR, SRR, IRR, I-HRR and p-HRR.

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 4.6: NDCG and RI values obtained in the experiments with CAS, CAS-or, NQE+m, HRR+m, SRR+m and IRR+m.

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

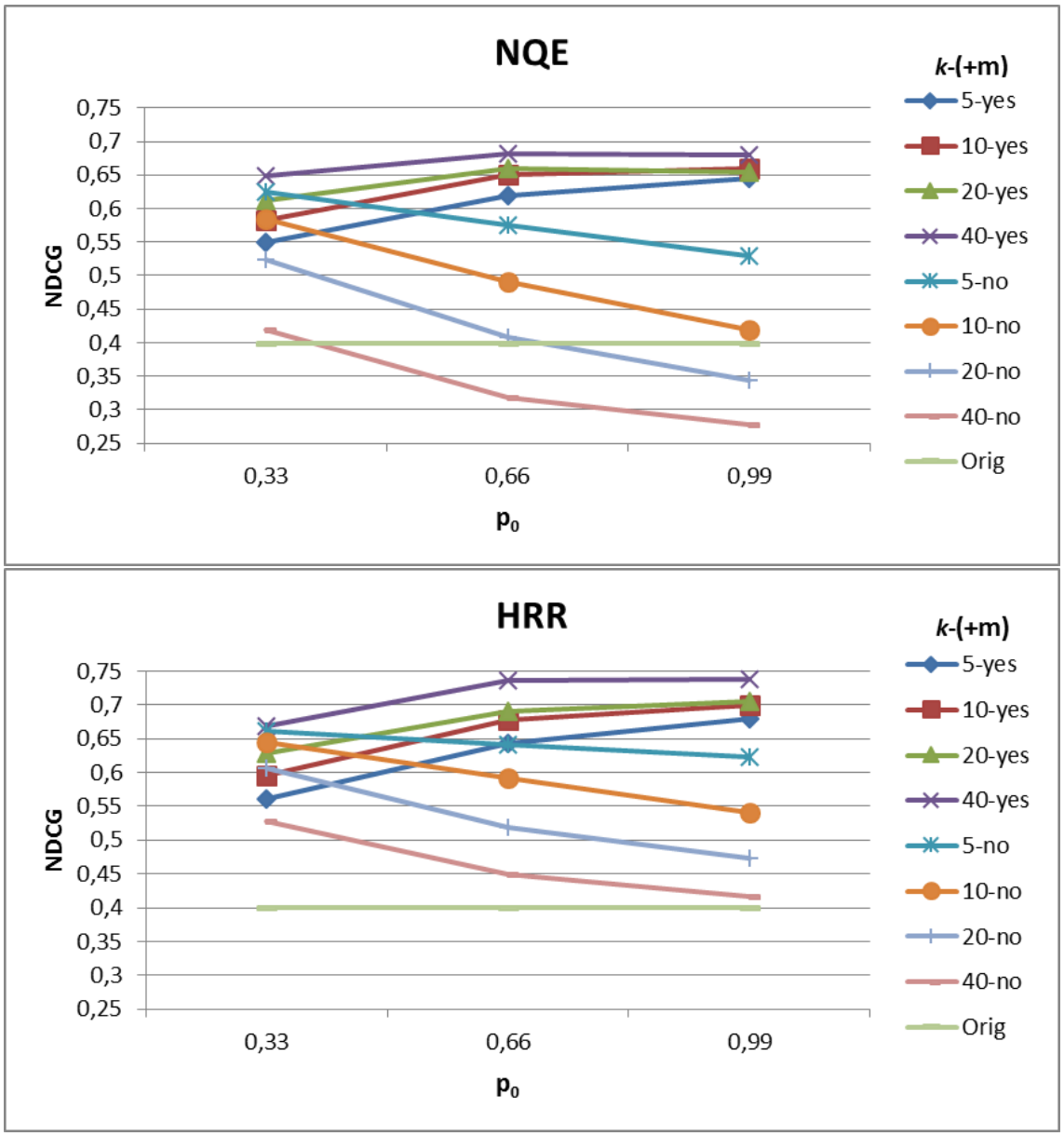

Figure 4.4: NQE(+m) and HRR(+m) NDCG values from Tables 4.5 and 4.6. In the legends, the *yes* and *no* indicates whether the modification of the retrieval model has been used or not, respectively.

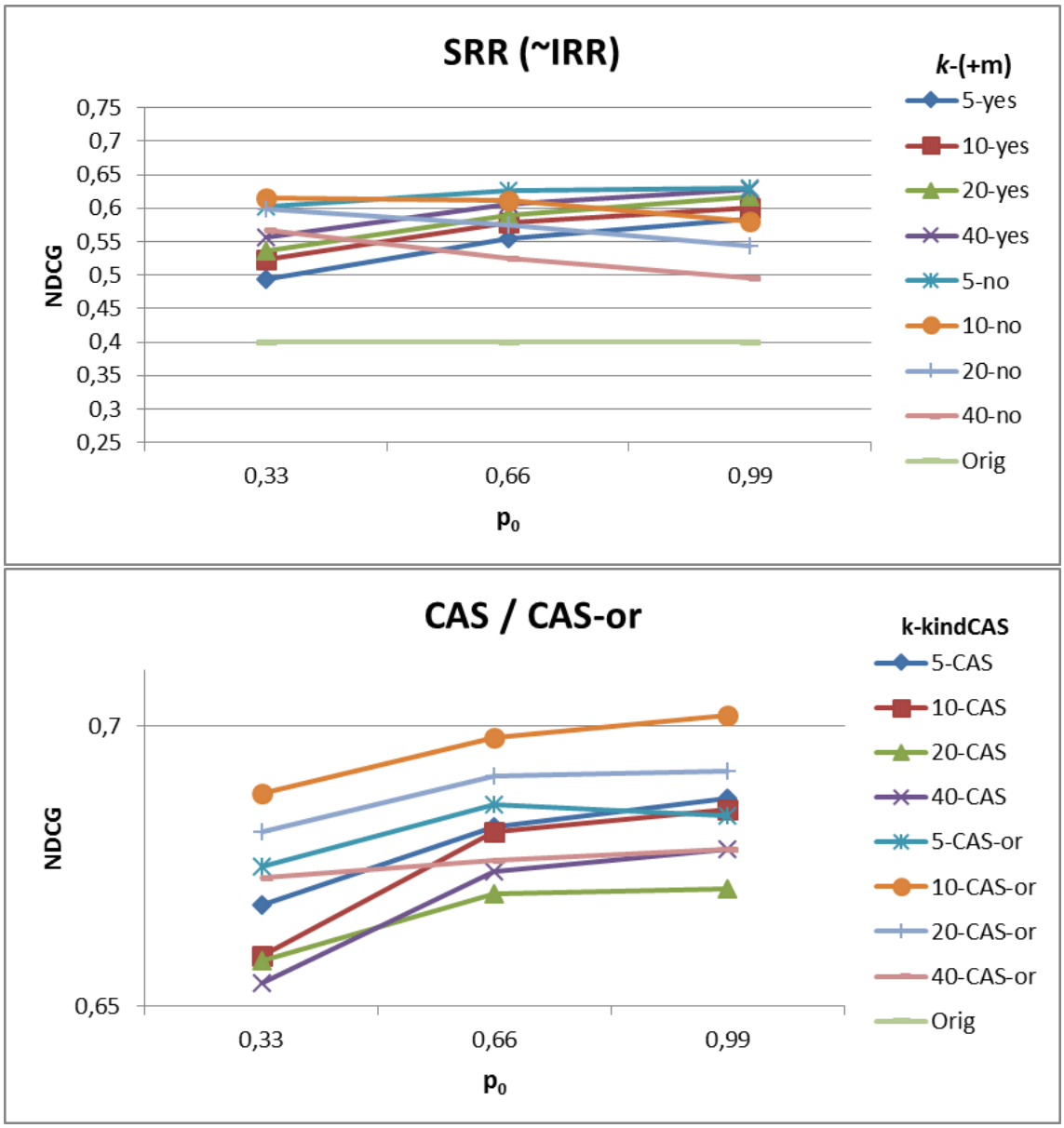

Figure 4.5: Same as Figure 4.4 but for SRR(+m)/≈IRR(+m) and *CAS*/*CAS-or* NDCG values.

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 the experiments, among which the following stand out:

- *QE* heavily depends 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, as can be seen in Figure 4.4, 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 $k = 5$ 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 and Figure 4.5 show). SRR and IRR also obtain much better values of RI than HRR, so that their behaviour is more robust across different queries. As their own name and design suggest, soft reranking (SRR) and IRR (based on it) smooth the results (see the same previous Figure 4.5).

- *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. This fact shows that our reranking personalization techniques design is the proper one.

- The use of *CAS queries* produces very good results, which are better than the corresponding results of all the previous strategies for most user profile configurations. 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 behaviour can be seen in Figure 4.5, where a new scale is necessary to be able to see the different user profile configuration lines for the CAS queries. 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). Thus, 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.

- In the case of *NQE+m, HRR+m, SRR+m* and *IRR+m* personalization strategies, the relation between the number of profile terms and the 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. These techniques disable the *query-drift* problem, as now the more terms and normalization factor, the higher the obtained performance, which can be perfectly seen in Figure 4.4.

  Additionally, 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 case 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 4.7 and 4.8 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 4.7 are slightly worse than those in Table 4.5, and the opposite is true for Tables 4.8 and 4.6. 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

Table 4.7: 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 4.8: 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

it (in the same way we have done with Garnata) in order to test its combination with the HRR 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 HRR if the number of terms in the profile is low (ten or less), and SRR otherwise (in both cases with a low p_0 value).

4.6 Conclusions and future work

In this chapter 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 techniques which exploit these profiles, in order to offer to the users those parts of XML documents that better reflect their interests and 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. Although these techniques have been developed for XML retrieval, all of them (except the structural query expansion) may be easily adapted to work with plain documents.

We have experimentally tested our personalization strategies, by means of a user study on a parliamentary document collection marked up in XML, aiming to mea-

sure the benefits of managing the user profiles with these strategies. 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 a NDCG improvement of 75.5% (71.25% on average) and 84.5% (67.25% on average), respectively.

As future work, we would like to study how to best select or tune the configuration parameters (i.e. the number of terms from the profile to use and the normalization factor of their weights), for example depending on the characteristics of the query, in order to obtain better personalized results. We also want to study the way of using any other contextual information included in the profile (other than search terms), as for example the gender and age of the user. Other possibilities are to include some novelty or diversity results to do not focus too much in the current user profile state, and allow the user to discover new information, or even to predict in some way the future user interests. Although in this chapter we have focused on the personalization of the content part of the XML retrieval, another interesting approach would be to also consider the structural part, e.g., how to personalize CAS queries or how to manage structural preferences.

Chapter 5

An Automatic Evaluation Framework for Personalized IRSs

5.1 Introduction

The evaluation step is very important for any developed system, since this process measures whether or not the system meets its original objectives, and how good it performs the task it was designed for, i.e., the system performance. In our particular case, where personalization services are becoming almost essential, in order to find relevant information tailored to each individual or group of people with common interests, it is very important to be able to build efficient and robust personalization techniques to be part of these services. For this reason, the evaluation step is a crucial stage in their development and improvement, so much more research is needed to develop easier, faster and robust personalized evaluation frameworks, being precisely that the objective of this chapter.

Any personalized system is composed by three main different stages: 1) how to acquire and represent the information about the user, which will be stored in the user profile, 2) how to exploit this user profile information in order to retrieve the most relevant results, which satisfy the user information needs, and 3) how to evaluate the whole personalization process. This chapter is concretely focused on the evaluation of the second personalization stage.

The evaluation of personalized IRSs is very difficult due to their complexity, since usually there is an underlying implemented personalization method, which tends to

have many configuration parameters to be adjusted, and subjective components as relevance assessments come into play, between other issues. For these reasons, and the involved potential costs of evaluation, most personalized IRSs do not conduct real world experiments for their validation [140]. However, this kind of experiments would be necessary in order to show the real IRS effectiveness, and to discern whether this given IRS provides or not improvements over other IRSs.

To evaluate traditional system-centred IRSs, where the user is not an integral part of the retrieval process, an evaluation framework based on the Cranfield paradigm [28] is normally used. This evaluation methodology aims to ensure *repeatable* and controlled experiments between different IRSs, extracting *comparable* measures and *generalizable* conclusions about them. System-centred evaluation methodologies have contributed to have a very good performance of general IRSs nowadays. But looking to this approach from a more practical point of view, it is actually very limited, because it does not consider anything about the context of the IRS final real users. IRSs under this approach are not able to adapt their results to their users, whose activities are complex and subjective by nature [12].

The word *context* is defined in [1] as: "Any information that can be used to characterize the situation of an entity, where an entity can be a person, place, or physical or computational object". Under our methodology we always refer to context as the user interests and preferences.

Pursuing the inclusion of the user context into the evaluation process, several user-centred evaluation frameworks have been developed. They can be classified into three main categories, which have been previously explained in Section 2.4: extensions to the laboratory-based Cranfield paradigm, contextual simulations and the obvious user studies.

Actually, as it is stated in [43], it is not only about developing user-centred IRSs and the corresponding evaluation frameworks, forgetting the system-centred side, but to combine both evaluation perspectives: get the best of the system-centred approximation, adapting it in order to also get the best final user satisfaction in their daily experience with the system.

With the previous intention in mind, we have developed an automatic evaluation methodology to evaluate the retrieval effectiveness of any personalized IRS. This methodology will be denoted as **ASPIRE**, acronym of "Automatic Strategy

for Personalized Information Retrieval systems Evaluation". ASPIRE combines the repeatable, comparable and generalizable main advantages of system-centred approaches, together with the inclusion of the user context into the evaluation of the retrieval process, which is the main benefit of user-centred approaches. At the same time, ASPIRE avoids the interaction with real users, which is the cause of the user studies evaluation lack of control, not repeatable, not comparable and not generalizable results. ASPIRE also allows to avoid the difficulty and big associated costs of congregating several users, in some cases experts, for a long time to perform the evaluation process, including questionnaires, interviews, etc. The combination of the system-centred and user-centred evaluation frameworks advantages allows a fast and easy testing of any personalized IRS.

Thus, given two or more personalization approaches, one of the main *goals* of our methodology is to reach a ranking of them that is close to the one that would have been obtained by the same methods using real user interactions. In a similar way, another ASPIRE goal is to be used to set up the best configuration parameters for a given personalization technique. To accomplish both goals, the use of traditional evaluation frameworks is very difficult, or most of the times impossible, mainly due to the required user effort, in terms of time spent making personalized judgements over a set of topics (the assessments vary with the users). However, these judgements represent a key component to obtain the desired ranking. The use of our proposed methodology turns this process into an easy, low effort and low cost task.

As any other system or methodology, ASPIRE will be also evaluated in order to check its validity, reliability and robustness for the evaluation of personalized IRSs. ASPIRE mainly tries to be an alternative to user studies which, if performed in a controlled way, are the best real user-centred evaluation methodology. Therefore, we compare its results against the results obtained from a real user study.

It should be clear that ASPIRE does not pretend to completely replace user studies, rather it should be considered as an alternative to them. Although ASPIRE joins the advantages from both system-centred and user-centred evaluation approaches, it is very important to collect qualitative information about the IRS from real users [136]. ASPIRE pretends to be an alternative to user studies for the evaluation of personalized IRSs, specially in their first development stages or when the user study is not feasible due to any circumstances, such as the lack of resources

or time. ASPIRE also helps to make final user studies experimentation more worthwhile, by limiting the number of personalized IRS configurations that users must evaluate.

The remainder of the chapter is structured as follows. Section 5.2 describes our proposed automatic evaluation methodology, ASPIRE. Section 5.3 gives an overview of the different IR evaluation strategies existing in the literature. Section 5.4 shows ASPIRE use and validation through the definition of some metrics, and Section 5.5 shows the comparison of ASPIRE results with those obtained from a user study and with a state-of-the-art approach, using several retrieval models. Finally, Section 5.6 finishes with some general conclusions and proposals for future work.

5.2 ASPIRE

There are several methodologies for the evaluation of personalized IRSs (see Section 2.4), but there is still no agreement about the definition of a standard evaluation framework and the evaluation metrics to be used, since all the previous methodologies have different disadvantages. As the issue of evaluating personalization is significant and the evaluation of personalized systems is a crucial stage in their development and improvement, much more research is necessary to overcome this issue.

Our proposed ASPIRE methodology (Automatic Strategy for Personalized Information Retrieval systems Evaluation) aims to join the advantages of system-centred and user-centred evaluation approaches. In particular, ASPIRE produces repeatable, comparable and generalizable results (main benefits of system-centred approaches), which allows an iterative evaluation process, which in turn, lets a fast and easy IRSs development. At the same time, ASPIRE is designed to evaluate different personalized techniques, which allows the integration of the user context within the retrieval process (main benefit of user-centred approaches). ASPIRE evaluation results offer a compromise among the quantitative and controlled results of system-centred approaches and the qualitative results of user studies.

Since ASPIRE is an automatic evaluation methodology, where no interaction with real users is required, it is framed within the personalized evaluation category of contextual simulations. ASPIRE shares a lot of features with contextual simulations,

but at the same time, it has some features which facilitate its execution among other contextual simulation approaches. ASPIRE evaluation results meet the compromise between the advantages of system-centred and user-centred approaches.

ASPIRE pretends to be a configurable tool which allows to select the best personalized approach between any two or more number of them, according to the selected evaluation metric, or even to select, within the same personalization approach, its best configuration parameters set up. Additionally, it transforms all this process into an automatic process, selecting the best personalized approach with low effort and cost. This last feature represents an important advantage against other personalized evaluation frameworks, especially those where real users are involved, but also with respect to the own contextual simulations it belongs to, where an exhaustive well defined retrieval scenario must be specified in order to simulate users and user-system interactions.

We next detail all ASPIRE evaluation framework components:

- **Document collection**: we can use any document collection, with the only requisite that the documents in this collection, or at least a subset of them, must be *able to be classified* into different areas of interest or categories. This classification could be explicitly performed by another system component, or the own document collection may be already implicitly classified. An explicit example would be that the documents had some associated tags (e.g. from a controlled vocabulary), either manually or automatically assigned. In the case of this last approach, a first step based on a clustering process could be used to find clusters of similar documents, according to their contents. Later a classification process could assign new documents to the corresponding clusters. An implicit example would be that the documents were already classified by its own nature, such as a newspaper, where its news are classified into sections, which represent its different areas of interest or categories (e.g. sports, international, ...), or the case of the document collection we have used in our experiments, where each one belongs to a specific parliamentary committee (economy, agriculture, employment, ...).

- **User Profiles**: users are simulated, *no real user* interaction with the IRS is required. Each of these simulated users is associated with one or more areas

of interests of the document collection. Consequently, we assume that each simulated user will be interested in the topics of the documents which compose the selected area(s) of interest. There are several ways of representing the interests of a (simulated) user by means of a user profile. The most common is to use a set of weighted terms. We use this strategy, extracting the set of profile terms from the content of the documents corresponding to the area(s) of interest associated to the simulated user. This can be done by means of an automatic learning process of the most representative terms of these documents (based, for example, on term frequency (tf) and/or inverted document frequency (idf)).

- **Queries**: *any query* may be used, although we advise to use queries formulated by real users of the document collection (e.g. obtained from a log file). An heterogeneous set of queries should be used for better retrieval evaluation, representing a trustworthy sample of real user information needs.

- **Relevance assessments**: one of the main drawbacks of almost all evaluation systems is the need of having previously assigned relevance assessments for each query, or to have real users judging the relevance degree of documents for a given query. ASPIRE avoids this problem by using a process that *simulates the relevance assessments* (documents which are relevant for a given query and a given user profile). We do it in the following way: we run the given query against the non personalized IRS obtaining a ranked list of results. A document will be considered relevant for a given user profile (and the query), if it belongs to the area(s) of interest this user profile represents, and it has been retrieved by the IRS among the first *topkRel* results. The intuition behind this procedure is: if a document is among the first ones retrieved by the system for the query and it also belongs to the area(s) of interest associated to the simulated user, then probably this document is simultaneously about the topic of the query and those of interest for the user. Hence, it is relevant for the query from the specific point of view of the simulated user. The parameter *topkRel* should be a relatively low threshold, because it would be really unusual for a real relevant document to appear in a very low position in the ranking, let us say, in the position 1000. If we use a high *topkRel* value, then we likely would

introduce many false positive relevant documents for the given query. We will see in Section 5.5.2 why *topkRel* is an important parameter for the relevance assessments criteria.

Any personalization technique can be evaluated, provided that is compatible with the user profile representation being considered. In this sense, any evaluation metric can be used, although we consider particularly valuable the use of rank-based evaluation metrics, as for example, NDCG which was designed to measure the quality of a given ranking, by computing the normalized relevance degree weighted sum of a given ranked list of documents. It assumes that highly relevant documents are more useful if they are on the top of the ranking, being at the same time more useful that marginally relevant documents.

An important parameter for the computation of NDCG is the number of results being considered for the evaluation of the system performance, denoted as *topkEval*. In some way *topkEval* represents the number of documents a user could evaluate during his/her interaction with the system. In this sense, we suggest that *topkRel* should be higher than *topkEval* threshold, i.e., $topkRel > topkEval$. By means of this fact, we give the opportunity to the personalization techniques to push up some potentially relevant results into the *topkEval* range.

The proposed ASPIRE evaluation framework, although being a contextual simulation, has some advantages over them. ASPIRE is mainly developed to test personalized IRSs retrieval effectiveness. Contextual simulations are more devoted to evaluate the interactions between the user and the IRS. This feature allows the evaluation of the IRSs interfaces effectiveness, but this is not the aim of ASPIRE. This extra capability of contextual simulations comes at an associated cost. They need a deep definition of the retrieval scenario, defined a priori by a sequence of user interactions with the IRS, which are not always easy to define. In contrast, ASPIRE only needs a classifiable document collection (as mentioned earlier, if the collection is not preclassified, a clustering process could be used to define the categories) and a set of common queries for this collection. That is to say, ASPIRE allows the evaluation and improvement of personalized IRSs with a very low effort under a completely automatic evaluation process.

5.3 Related work

This related work section focuses on approaches similar to our proposed ASPIRE evaluation methodology. For a more general overview about the state-of-the-art on evaluation research, for both system-centred and user-centred approaches, check Section 2.4.

There are not so many approaches similar to ASPIRE, but some in the same line. We next outline some of them in increasing order of similarity with respect to our approach.

One of the first approaches was [84], whose main purpose is to take into account the dynamic interests of users within the user modeling. The authors have developed a simulation-based information filtering system to overcome difficulties on studies where user factors, such as the environment conditions or their current mood, can impact interests. This system uses an approach known as reinforcement learning for user modeling. Some different scenarios are performed with this system to examine model accuracy and filtering effectiveness.

Another approach is [137], which evaluates relevance feedback algorithms using searcher simulations through different interfaces, with the intention of determining which of these models to use in the final version of the interface. The search interfaces provide interactions as a source of evidence for the models, using viewed results as the indicator of relevance. The searcher simulations allow them to have more controlled experimental conditions and to model complex interactions without the need of real users. Authors specify that the conclusions derived from their work are still provisional, since they use the evaluation methodology to evaluate implicit feedback models, but the methodology itself is not validated. This is exactly what we do in Sections 5.4 and 5.5, to validate our proposed methodology with a real user study.

From the two previous references, the first one is focused on user modeling improvement, and the second one on evaluating how search interfaces provide more relevant information to relevance feedback algorithms, both focusing on different aspects with respect to us. In addition, they must define the user-system interactions which will compose the retrieval scenario.

In [32], an evaluation protocol for session-based personalization (searching a sequence of related queries, i.e, short-term personalization) is proposed. The profiles are based on the topics provided by the TREC HARD collection. The user profile is simulated for each topic using a set of documents returned by the system, which have previously been judged as relevant by TREC assessors. The queries in a session are built by selecting the top terms associated to subtopics (subsets of relevant documents for the topic). The emphasis of the evaluation is put in the delimitation of the session boundaries. The main differences with our proposal, in addition to the focus on short-term personalization, are that they use simulated queries instead of real ones (as also done in [115]) and that relevance judgements are not simulated but real.

The closest approach to ASPIRE is proposed by Sieg et al. in [115]. The authors build user context models as ontological profiles, assigning implicitly learned interest scores to existing concepts of a domain ontology (ODP). As these interests are dynamic, a spreading algorithm is used to maintain them updated along time. Their aim is to demonstrate that reranking improves the disambiguation of the user query intent. They use a document collection of 10226 documents indexed under various ODP concepts, leading to three different sets of documents. The training set is used for an ontology representation (associating ontology concepts with sets of terms in the collection), the profile set is used for the spreading activation algorithm, and the test set is the document collection for searching. They automatically build four variations of keyword queries using the top terms associated to the concept/user interest being simulated. For each query, a new instance of the ontological user profile is created, performing the spreading activation algorithm to update user interests. The query results are retrieved using a cosine similarity measure for matching, each of those results being considered relevant if it is classified under the simulated concept, and not relevant otherwise. With the user profile, they rerank the original search results calculating again top-n recall and precision with the personalized results. Finally, they compare the original and personalized metric results, concluding that the reranking process improves the disambiguation of the query.

Although Sieg et al. approach [115] and our proposed methodology seems very similar, they actually have several differences, such as: 1) they focus more in the user profiles than in the retrieval effectiveness of the different personalized techniques;

2) they disambiguate ambiguous queries more than personalize them; 3) the way they build the queries based on an unique concept of the ODP ontology, which may not represent real user information needs, whereas we use real queries suitable to be evaluated under more than one document collection area of interest. And maybe the two most important differences: 1) we verify our evaluation methodology validity with a real user study, and 2) we design a generic and automatic evaluation methodology, with the intention to be easily used under very different evaluation situations for any personalized IRS approach.

There are also some interesting works, not focused on personalized but on general retrieval systems evaluation, which also propose methods without using real relevance judgements. In [89], a pool of documents is generated from the top b documents returned by each of the systems being evaluated for a given query. These documents are ranked according to their similarity with the query using a vector space model, and the top s documents of this ranking are assumed to be the (pseudo) relevant documents for the query. In [116] the (pseudo) relevant documents are extracted from the pool randomly, using a simple model for how relevant documents occur in the pool. In both cases the correlation between the rankings of retrieval systems using these simulated judgements and using human judgements was not very high (a Kendall τ correlation always below 0.5) in experiments with TREC collections.

In general, the stability of systems rankings is measured by using Kendall rank correlation, τ. We have to be cautious when the rankings are obtained over narrow score ranges [106], since low τ values might be expected. As [91] concludes, the evaluation strongly depends on a relative small set of top-ranked results. So, whenever our automatic methodology is able to find such kind of documents, we might expect to obtain rankings similar to those obtained within the user study.

Finally, although not directly related to our work, there is a set of studies which consider ranking evaluation with low cost [13, 17, 91, 13]. Some clear differences might be stated. The first one is that these papers are related to the reliability and robustness of relevance judgements to evaluate information retrieval systems. They focus on the number of queries, pool depth, etc. In these cases, all the relevance judgements are assumed to be true. Additionally, some works can be found which introduce error in the relevance assessments, studying how the systems rankings

were affected [116, 17]. But in all the cases, they do not tackle with the problem of personalization, in such a way that what is relevant for a user might not be relevant for another.

Other approach for low cost evaluation is the use of simulated queries using generation models that simulate a candidate query for a given set of documents, which are assumed to be relevant for that query [6, 46]. Although these papers reveal interesting trends, further studies in this direction are necessary in order to provide comparable results to manually assessed judgements.

5.4 ASPIRE use and validation

Our main objective is to validate the reliability of the proposed ASPIRE methodology for the evaluation of personalized IRSs. Moreover, we are going to show how ASPIRE allows to test and select the best personalization techniques from a set of different personalization techniques, also considering for each of them, the possible configuration parameters of the user profiles. This test and selection process is usually very difficult, or most of the times impossible, with traditional evaluation frameworks. However, the use of our proposed automatic evaluation methodology turns this action into a fast decision process.

In Chapter 4 we proposed and evaluated a wide set of 13 different and heterogeneous personalization techniques, using the relevance assessments from a user study. We are going to compare this study results with those obtained by applying ASPIRE, under the same circumstances and considering different retrieval models, in order to provide evidences about the reliability of our automatic evaluation methodology.

The different personalization techniques results using the Section 4.4.2 user study relevance assessments are considered as the real results, since we followed the advices given in [12, 24] for ensuring the reliability of this user study results. Therefore, the personalization techniques results evaluated under the ASPIRE evaluation framework should be close to them, or at least to follow the same dynamics (high correlation values), in order to validate ASPIRE.

5.4.1 Experimental framework

For the ASPIRE use and validation we have used an experimental framework composed by the following components.

Search engines. We will explore our approach using three retrieval models. On the one hand, our methodology will be tested with a search engine specifically designed for dealing with structured documents and, on the other hand, we will consider retrieval models designed for working with flat documents. The selected retrieval models are:

- *Garnata*: the specifically designed search engine and retrieval model designed to work with structured documents is Garnata (see Section 3.7). After submitting a query, the system ranks a set of non-overlapping elements according to their relevance to the topic.

- *BM25*: the second retrieval model is based on a probabilistic retrieval approach, particularly considering the BM25 term weighting formulas [102]. BM25 has been used quite widely and successfully across a range of collections and search tasks, representing a state-of-the-art tf-idf-like retrieval function.

- *VSM*: the third approach is a vector space retrieval model (VSM) [105]. The similarity function is derived from the classic cosine measure, which can also boost term weighting based on user specified requirements, e.g., the importance of the fields. Note that, since the data used in the evaluation only contain one field, we do not consider the boosting factors.

BM25 and VSM have been used under their corresponding implementation in the popular Lucene open-source search engine[1]. Lucene provides indexing and search technologies, which is frequently used by several applications all over the world, ranging from mobile devices to sites like Twitter, Apple and Wikipedia. This search engine is designed to work with plain (non-structured) documents.

Document collection. Since the used retrieval models work with structured and flat documents, we will consider two different versions of the records of parliamentary proceedings of the regional parliament of Andalusia (Spain). Just to

[1] http://lucene.apache.org/

remember it, each record corresponds to a given committee session, where each of these committees is dedicated to a specific area of interest, e.g., economy, health, education, employment, etc. Notice that for both collections each retrievable element only belongs to one committee. Therefore, the collection itself is already implicitly classified, as required by ASPIRE. The two different versions of the used document collection are the following:

- *Structured-XML collection*: it is explained in detail in Section 4.4.1. It has 658 XML documents with a total number of 432575 different retrievable elements (e.g. an intervention of a member of the parliament, or a paragraph within this intervention) with a size of 122MB.

- *The flat version of the previous collection*: it is obtained after considering one different document for each XML document initiative. Each of these obtained documents includes all the information relevant for the initiative, without any structure. In this case, we have a total number of 3732 documents (which gives an average of 5.67 initiatives per original xml document), with a similar size to the structured collection.

Relevance assessments. We have the relevance assessments from the carried out user study for the previous structured and flat document collections, as well as the relevance assessments automatically generated by ASPIRE for both collections. We next explain how these different sets of relevance assessments have been generated.

- *XML document collection*: the associated relevance assessments to this collection are exactly those generated in the carried out user study for its 126 evaluation triplets (see Section 4.4.2).

- *Flat document collection*: the previous XML document collection relevance assessments have been extrapolated for this flat document collection. Particularly, we have considered that an initiative (a document in this flat collection) will be relevant if itself or any of its descendants has been judged as relevant in the corresponding user study evaluation triplet.

- *ASPIRE automatically generated relevance assessments*: the same user study 126 evaluation triplets IRS results have been used to obtain the ASPIRE simulated relevance assessments for each of the used retrieval models (Garnata, BM25 and VSM). Briefly, we submit each original query to each retrieval system, focusing on the *topkRel* obtained highest ranking results. Then, each one of these results is considered relevant to this topic, if and only if it belongs to the same area(s) of interest than the one(s) represented by the user profile (see Section 5.2). Since in the user study the users evaluated up to 100 (50+50) results, we have fixed *topkRel* to 100 in our experiments.

5.4.2 Validation methodology

Notice that by using ASPIRE assessments we are considering as relevant documents some good candidates (they are highly ranked and belong to the same topic of interest) but, on the other hand, some errors are included in the assessments: some relevant documents may be missed (if they were considered as relevant by the user, although not belonging to the user profile area(s) of interest) and some non relevant documents might be considered as relevant (if they were considered as non relevant by the user, although belonging to the user profile area(s) of interest). Now, the problem is to measure the impact of these mistakes on the evaluation.

To tackle this problem we are going to consider two different criteria: on the one hand, we conduct a comparison between both (real and automatic) relevance assessments. By means of this comparison, we can measure the amount of error. On the other hand, we want to evaluate whether ASPIRE is (or not) a reliable methodology to evaluate personalized IRSs. In other words, if the errors made in the assessments could cause too much damage in the aggregate, which invalidates the conclusions obtained by our approach. In this sense, we will compare the retrieval performance (using the NDCG metric) obtained by the user study and by ASPIRE, under different personalization techniques and user profile configurations.

In this experimentation we have fixed the number of results being considered for the evaluation of the system performance, i.e. *topkEval*, to the first 50 results. Note that this value meets our requirements, since $topkEval = 50 < 100 = topkRel$. We next present the metrics employed for these purposes and then, the used personalization techniques and profile configurations.

Evaluation metrics. *Firstly*, we will focus on the comparison between the relevance assessments from the user study and from ASPIRE. To accomplish this goal we have considered two evaluation metrics.

- The first metric measures which percentage of the real relevance assessments associated to each evaluation triplet (*query, profile, user*) truly belongs to the documents within the corresponding user *profile*. Remember that the main assumption of ASPIRE is that the simulated relevance assessments are always extracted from the document collection area(s) of interest associated to the simulated user profile. Then, if an element in the real relevance assessments does not belong to the area(s) of interest the user profile represents, it will never be considered as a relevant result by ASPIRE, and always considered as relevant otherwise. In some sense, this metric measures the confidence we can expect from the ASPIRE relevance assessment criteria.

  We define this evaluation metric by the following formula:

$$\text{Conf}(q,p,u) = \text{size}(et_{q,p,u} \cap id_p)\frac{100}{size(et_{q,p,u})}, \tag{5.1}$$

  where $et_{q,p,u}$ is the set of relevance assessments of the evaluation triplet for the given query q, profile p, and user u; and id_p is the set of all documents in the collection belonging to the profile p.

- The second evaluation metric aims to measure the overlap degree between the relevance assessments provided by the user study and by ASPIRE. To do that, we propose to use the counterparts of the classical measures of precision (*pre*), recall (*rec*) and F from the classification field. Let $et_{q,p,u}$ be as in eq.(5.1) and $et_{q,p}$ be the set of simulated relevance assessments for the query q and the profile p.

$$pre = \frac{tp}{tp+fp}, \quad rec = \frac{tp}{tp+fn}, \quad F = \frac{2*pre*rec}{pre+rec}, \tag{5.2}$$

  where tp denotes true positives (number of relevance assessments in $et_{q,p}$ which are also in $et_{q,p,u}$), fp denotes false positives (number of relevance assessments

in $et_{q,p}$ which are not in $et_{q,p,u}$), and fn stands for false negatives (number of relevance assessments in $et_{q,p,u}$ which are not in $et_{q,p}$).

For XML documents things are a bit difficult because the elements in the sets $et_{q,p,u}$ and $et_{q,p}$ can match partially (the two elements are not identical but one contains the other). For that reason we need to use a function, $sim(A_j, U_i)$, measuring the degree of similarity between an element $A_j \in et_{q,p}$ and other element $U_i \in et_{q,p,u}$. This function is based on the distance between these elements in the XML hierarchy. Obviously if there is no overlap between A_j and U_i the similarity is zero (see 4.5.1 for more details).

Taking this similarity measure into account, the definition of tp, fp and fn in Equation 5.2 is:

$$tp = \sum_{sim(A_j,U_i)\neq 0} sim(A_j, U_i) \tag{5.3}$$

$$fp = size(\{A_j \in et_{q,p} \mid sim(A_j, U_i) = 0 \; \forall U_i \in et_{q,p,u}\}) \tag{5.4}$$

$$fn = size(\{U_i \in et_{q,p,u} \mid sim(A_j, U_i) = 0 \; \forall A_j \in et_{q,p}\}) \tag{5.5}$$

Secondly, and in order to measure the retrieval performance of a retrieval run (e.g. any personalization technique) we will use the NDCG metric, which has been proved valuable for the comparison of retrieval performance between systems [103].

When we work with the collection exploiting the XML organization, some NDCG adaptations are required to deal with partial matchings between retrieved elements and relevance judgements (see again Section 4.5.1 for details). For the flat document collection there is no need to make any modification in the NDCG formula.

Personalization techniques and profile configurations. One of the motivations of our methodology is to be able to obtain a ranking of personalized systems and/or configuration parameters for a given personalization technique, with the goal of selecting the best ones. In this sense, we ask the following question: how much would the rankings change if relevance assessments were chosen using ASPIRE instead of the real user study ones? In this section we will highlight the different alternatives for personalization analysed in our experimentation.

With respect to the ranking of personalized search, we have tested the structured-based and flat-based retrieval models, considering their behaviour under the original query (non-personalized), denoted as (*Orig*), and also the behaviour under a set of 13 and 7 different personalization approaches, respectively. In the last case, since BM25 and VSM are flat-based, we have not considered those XML-oriented alternatives, neither those that require a modification of the search engine, as we will explain later.

Within this wide set of personalization techniques (explained in detail in Section 4.3), there are approaches from the three possible retrieval stages where personalization can be applied: before the search (e.g. query expansion - *QE and NQE*), within the search (not very used yet, e.g. retrieval model modification - *NQE+m, HRR+m, SRR+m and IRR+m*), and after the search is performed (e.g. reranking - *HRR, SRR and IRR*). We have even included two bad performance personalization techniques (*I-HRR and p-HRR*), which are used to demonstrate some of the other personalization techniques design decisions, and we have still considered them here to support and strength the ASPIRE reliability. Finally, we have also included two additional content-and-structure personalization techniques (*CAS and CAS-or*).

As the reader may realize, several of the previous personalization techniques are hybridizations between some of the explained three basic approaches where personalization may be applied. We must recall that both the content-and-structure and within-the-search personalization techniques can not be used by BM25 and VSM retrieval models. We also have to note that more important than the specific characteristics of each of the used personalization techniques, is the fact that they cover a great variety of personalization approaches and different performances.

All of the 13 previous personalization techniques have an underlying common feature: in one way or another, all of them make use of an expanded query, which uses the appropriate user profile weighted terms in the expansion step. Based on this characteristic, we have tested all the personalization techniques under 12 different combinations of the two main user profile configuration parameters: the number k of used expanded terms (5, 10, 20 and 40) and a normalization factor p_0 applied over their associated weights (0.33, 0.66 and 0.99), which controls the importance of the expanded terms with respect to the original query terms.

Therefore, we have a set of 149 different IRS configurations to be tested ($(12 * 13) - 8(QE) + 1(Orig)$) under Garnata and 77 different IRS configurations to be tested ($(12 * 7) - 8(QE) + 1(Orig)$) under each of the BM25 and VSM models. Each of these IRS configurations involves the use of the 126 evaluation triplets, which represents a total number of 18774 and 9702 different ranked lists of retrieved results for each evaluation approach (user and ASPIRE) ready to be evaluated using NDCG. We will only display results based on the averages of NDCG values across the 126 evaluation triplets for the 149 and 77 different IRS configurations, respectively.

5.5 Results

In this section we will first show the comparison between the user study and ASPIRE results with our proposed evaluation framework using Garnata, BM25 and VSM retrieval models. Then, we will show the same comparison results between the user study and a state-of-the-art evaluation approach described in [115]. All these comparisons will show how reliable and robust is ASPIRE under different scenarios.

5.5.1 User study-ASPIRE results comparison

In this first part we will show both the relevance assessments and the retrieval performance evaluation comparisons, under our proposed evaluation framework. The relevance assessments comparison will try to validate the ASPIRE relevance criteria, while the retrieval performance comparison will show how well ASPIRE behaves with respect to the real user study evaluation results.

Relevance assessments comparison (validating ASPIRE relevance criteria)

In this case, we will focus on the user study obtained assessments using the XML-based retrieval model, where a deeper analysis can be done. Nevertheless, similar results would be obtained when considering the unstructured retrieval systems.

We will begin this point comparing the raw number of relevant results per triplet, see Figure 5.1, in the real user study (x axis) and the ones obtained using ASPIRE (y axis). From this graph, we can see that in general the automatic assessments tend

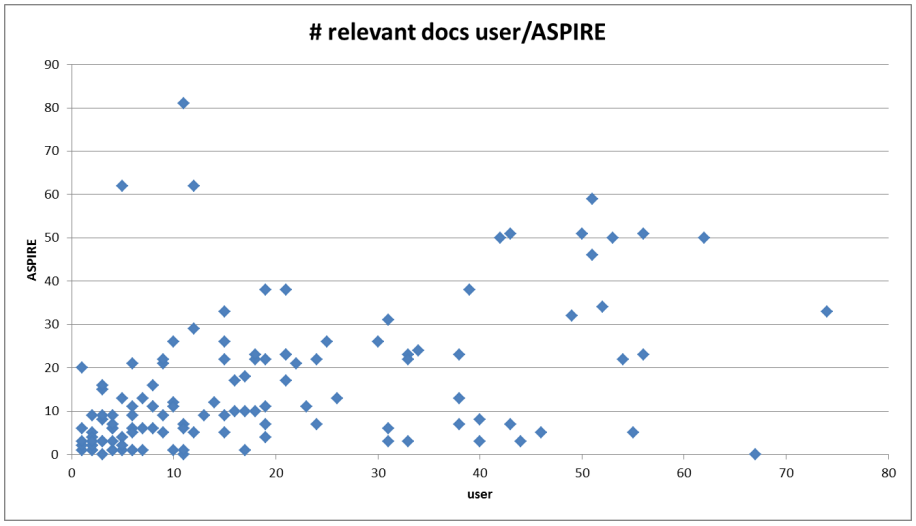

Figure 5.1: Number of user study evaluation triplets relevance assessments (x axis) against ASPIRE evaluation triplets relevance assessments (y axis). Each point in the graph represents an evaluation triplet.

to select a lower number of relevant elements (1965) than the user study (2362), although they are positively correlated with a Pearson correlation value of 0.452.

The application of the *Conf* metric (Equation 5.1) over the 126 evaluation triplets gives an average value of 75.93% with a standard deviation of 28.97%. This number means that approximately 3 out of 4 results judged as relevant by real users are also considered as potentially relevant by ASPIRE, conforming its main assumption about relevance assessments. We believe that this is a quite good ratio. Anyway, we will see later whether the remaining 24.07% of miss-assessed relevant results will cause a proportional difference when analysing the user study-ASPIRE retrieval performance comparison or not.

The average results, across all the evaluation triplets, of precision, recall and F measure, together with the standard deviations, are displayed in Table 5.1. We can observe large deviations, so that the behaviour is quite different depending on the query and the profile being evaluated. The average values indicate that the overlap degree between real and simulated relevance assessments is around 50%. As with the previous metric, the interesting question now is whether this overlap degree is good enough to produce sufficiently close performance evaluation results for both the user study and ASPIRE approaches.

Table 5.1: Averages and standard deviations of precision, recall and F metrics across the 126 evaluation triplets.

	pre	**rec**	**F**
μ	0.548	0.513	0.450
σ	0.307	0.334	0.260

Retrieval performance evaluation results comparison

Although the previous metrics are interesting, giving a first insight of the ASPIRE quality for the generation of relevance assessments, the actually important values are those based on the comparison between the evaluation of the retrieved results using the user study and ASPIRE. These results are the ones the final users will use to judge the behaviour of any personalized IRS, and if ASPIRE is able to evaluate them pretty similar to the way the users would do it, independently of the underlying relevance assessments, ASPIRE will be a good method to simulate those users.

It is important to remark that we have evaluated a very heterogeneous set of personalization techniques, ranging from some very good to some very bad performance strategies under different retrieval models. This comprehensive evaluation makes the derived conclusions more robust and valuable.

Here, we pursue two main objectives: 1) to test whether ASPIRE should be considered as a reliable approach in the evaluation of personalized IRSs, comparing its evaluation results with those obtained from the carried out user study. And, 2) to show whether ASPIRE is able to rank properly the personalization approaches, and even to rank the different profile representations for the given personalization methods, in accordance with the user study results.

Is ASPIRE a reliable evaluation approach?

To answer this question we will compare the performance for the 126 evaluation triplets for each configuration of personalization techniques and profile parameters, computing the average NDCG values for each run. As measure of the quality of our approach we consider whether the NDCG values obtained from the automatic evaluation might correlate (or not) with their respective values in the user study.

Table 5.2: NDCG user study-ASPIRE Pearson correlation ranges, average values and standard deviations for the different runs.

Model	Range	Avg	Sdv
XML	$[0.450, 0.716]$	0.604	0.057
BM25	$[0.177, 0.794]$	0.631	0.144
VSM	$[0.313, 0.797]$	0.591	0.137

We will examine these values at two different levels of granularity. Firstly, we consider each personalization technique-profile parameters combination as an independent run and compute its averaged quality over the 126 evaluation triplets (using NDCG) for both, the user-study and ASPIRE approaches. Then, we use Pearson correlations between the NDCG values of each run to determine the resemblance between the different results.

Table 5.2 and Figure 5.2 present the obtained results. Table 5.2 shows the Pearson correlations between the NDCG obtained with the user study vs. the ones obtained using ASPIRE. For illustrative purposes, Figure 5.2 shows the histogram which summarizes the previous table data for the XML-based retrieval model. In all the cases, we can observe a positive correlation with an average value around 0.6 with low standard deviations. Thus, ASPIRE can be considered as a moderately good predictor of the relative performance on individual evaluation triplets for the considered models.

The same correlation coefficients are plotted in Figure 5.3 (in the y axis) against the averaged NDCG values from the user study (in the x axis). The main conclusion drawn observing this figure is that the correlation values do not depend on the real (good or bad) performance of the given personalization techniques-user profile configuration parameters, except for a small number of outliers (for low Pearson correlation values).

The previous results represent an intra-configuration (personalization technique-user profile approach) comparison. Now, we will focus on how these configurations relate to each other. In this case, we consider an evaluation matrix where the columns represent the personalization strategies and the rows represent the profile representation parameters. In this matrix, each cell represents the average NDCG obtained

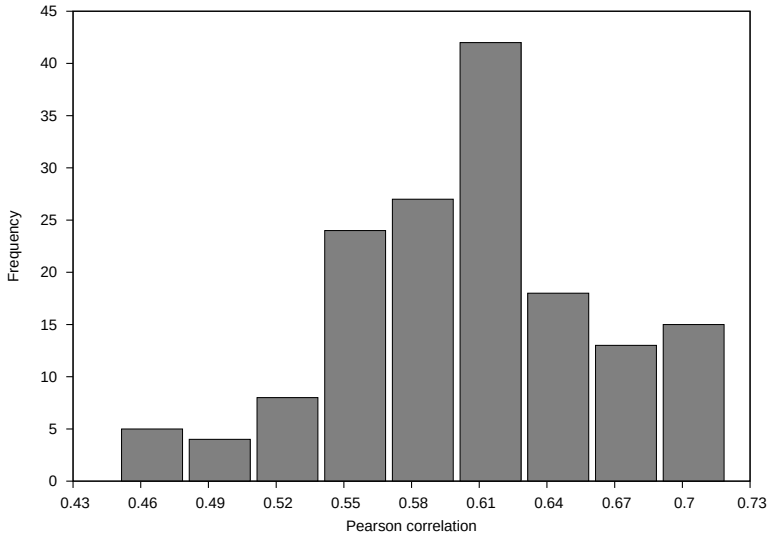

Figure 5.2: Histogram for Table 5.2 XML-based correlation values approach.

after running the 126 evaluation triplets under a given configuration, being a global measure about its quality.

Figures 5.4 and 5.5 plots the user study averaged NDCG values (x axis) against the corresponding values obtained from ASPIRE (y axis), for the 149 evaluation matrix cells for the XML-based approach and the 77 combinations of the BM25 and VSM retrieval models. In these figures, we also show the lineal regression line and the ideal fit line, labelled as *LR ($y = x$)*. Some conclusions may be drawn from these figures:

- They show how ASPIRE results are almost always very close to those obtained from the user study. A linear regression over this data shows a *R-squared* values of 0.8357, 0.945 and 0.940 for XML, BM25 and VSM, respectively. So, we have obtained very high *Pearson correlation* coefficients equal to 0.914, 0.972 and to 0.970 for each respective model[2].

[2]In this case the correlation values are higher than those in Figure 5.2, because here we are correlating the averaged NDCG values for ASPIRE and the user study, not the underlying and more diverse 126 evaluation triplets of each of these combinations.

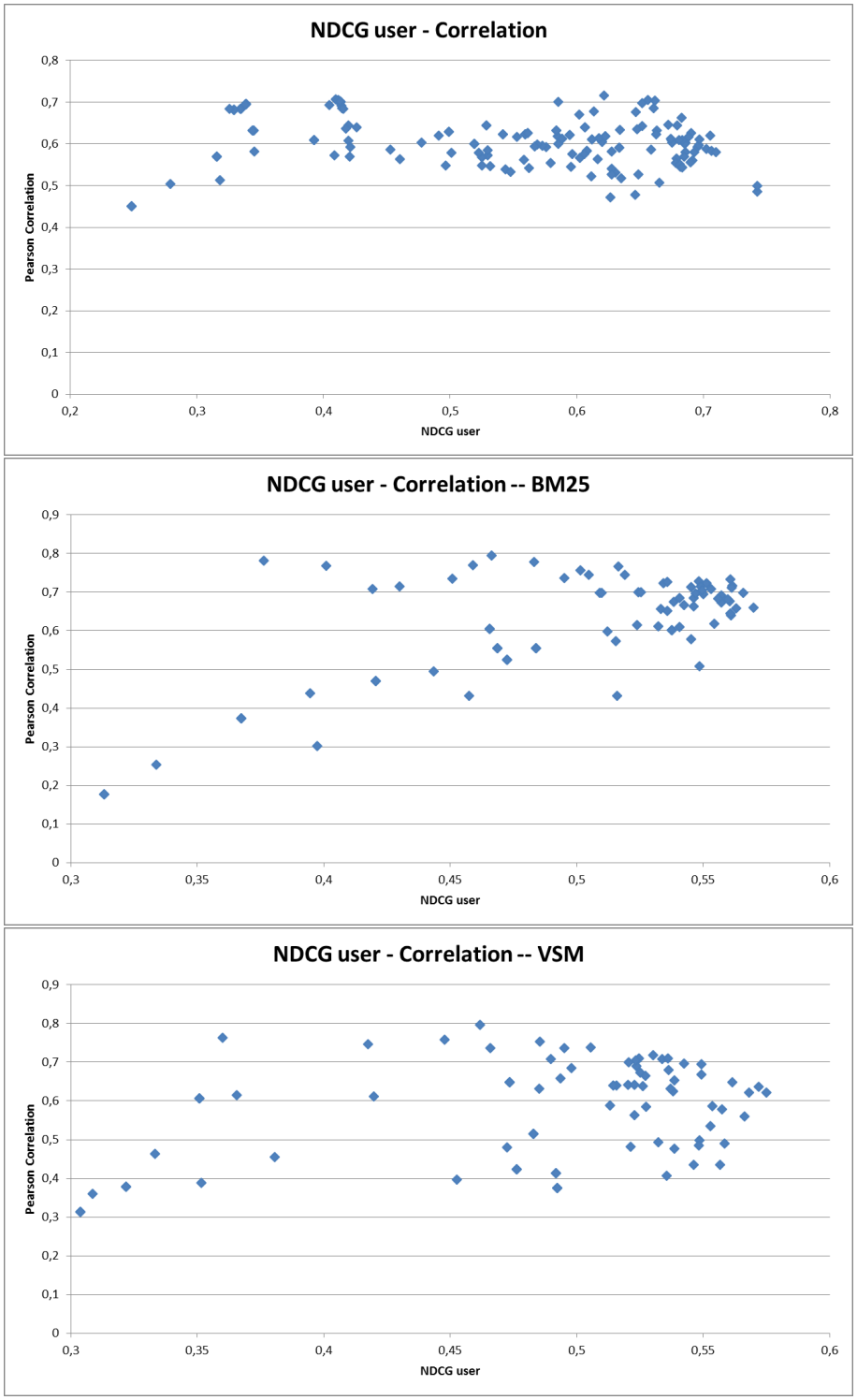

Figure 5.3: NDCG user study-ASPIRE correlations (y axis) against the user study averaged NDCG values (x axis) for the Garnata (XML), BM25 and VSM results, respectively.

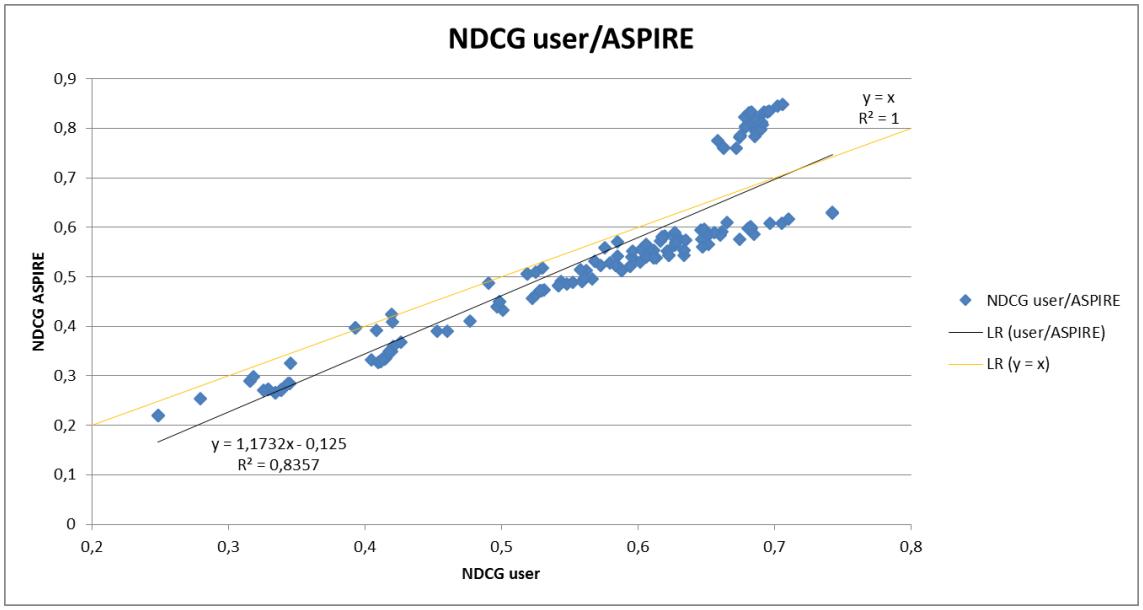

Figure 5.4: Averaged NDCG values from ASPIRE (y axis) and the user study (x axis), for each personalization technique-user profile configuration for the Garnata (XML) results.

- Considering the ideal fit line, labelled as *LR (y = x)*, we can show that ASPIRE results are very close to the real NDCG values (obtained from the user study). This is important because ASPIRE usually does not overestimate neither underestimate significantly the performance of the given personalization technique, independently of the retrieval model considered.

Taking into account all the results, we could conclude that ASPIRE is able to *robustly* evaluate any given personalization technique, independently of the used retrieval model.

Stability of rankings of user study-ASPIRE personalization techniques and user profile configurations

We are going to test whether we may trust on the systems ranking provided by ASPIRE, comparing it with that provided by the user study. Our objective is to look at the stability of rankings, rather than the absolute values of the NDCG metric. We use the Kendall τ rank correlation, also known as Kendall coefficient, to

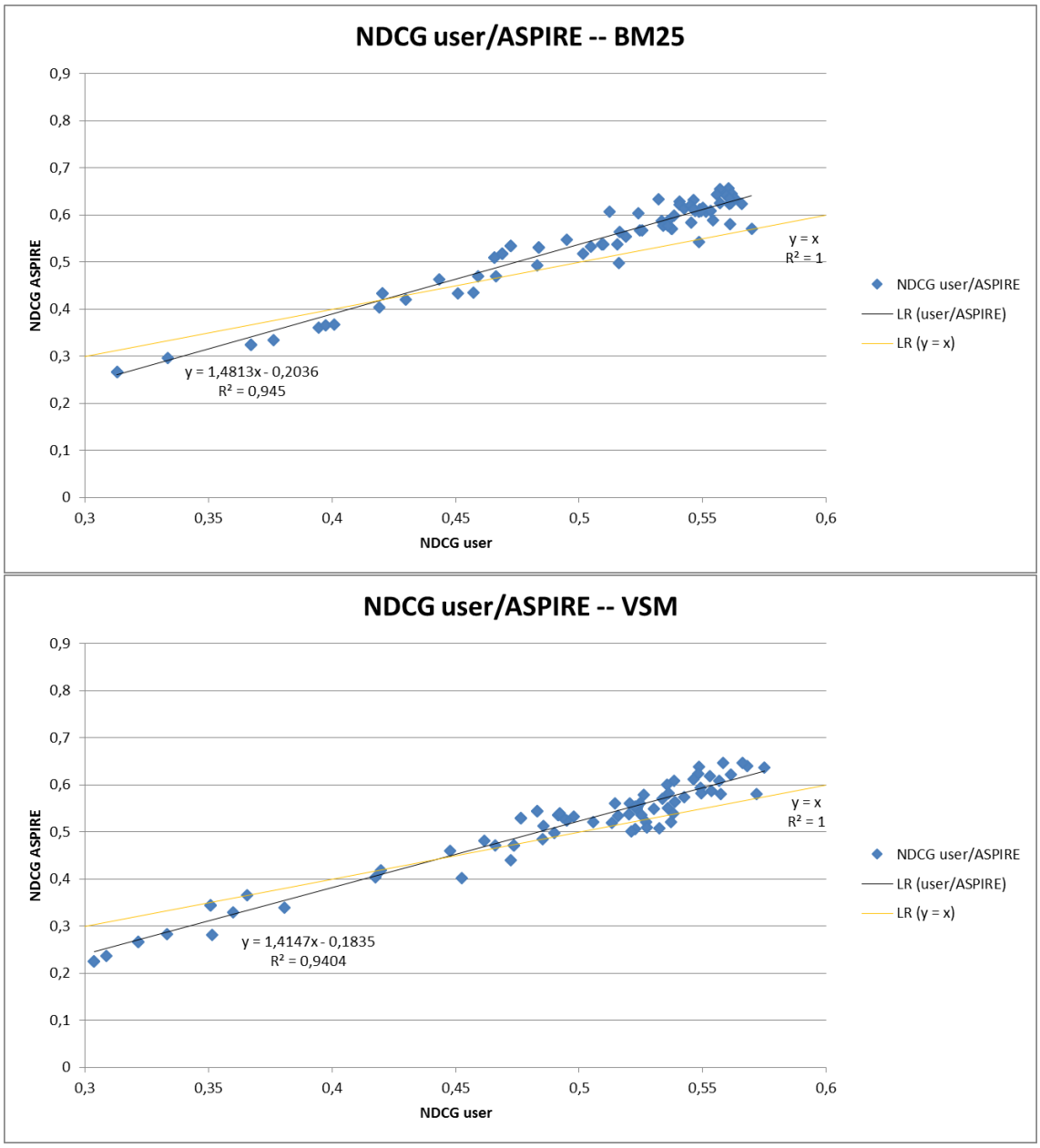

Figure 5.5: Averaged NDCG values from ASPIRE (y axis) and the user study (x axis), for each personalization technique-user profile configuration for the BM25 and VSM results.

compare both rankings. Kendall correlation is a function of the minimum number of pair-wise swaps required to turn one ranking into another. If the agreement between the two rankings is perfect, then $\tau = 1$; if the disagreement is perfect, then $\tau = -1$, and if both rankings are independent then $\tau \simeq 0$.

Taking into account that people usually disagree about relevance [108], problem aggravated in the case of personalized information retrieval, and that error in the assessments could affect the systems ranking [9, 18], it becomes necessary to contextualize the correlation value between the automatic and human-based evaluation. With this objective in mind, we also present the comparison results of the rankings obtained by considering two different user-based sets of assessments. These sets have been obtained after randomly split our original set into two independent groups, A and B. Then, by comparing the systems ranking for these groups, we might identify in which way the human differences in judgements may impact the relative system performance. This value, denoted as $\tau_{A/B}$, shall be used as reference for our comparison.

Selecting the best personalization techniques for a fixed profile

As we want to select the best personalization techniques, we are going to compute the rank correlations (Kendall τ) between the real and simulated results of the personalization techniques, for each of the 12 user profile configurations, displayed in Table 5.3. In this table, we also present the standard deviation among the NDCG values obtained for each personalization approach for both, the user study σ_{us} and the results obtained using ASPIRE, σ_{ASP}. These deviations are displayed to illustrate the differences in performance between the different personalization techniques: the higher standard deviations the higher is the difference between the performance of the different approaches.

In average, a Kendall $\tau_{XML} = 0.896$ with a low standard deviation $\sigma_{XML} = 0.063$ is obtained from the 12 user profile configurations for the XML-based model. Note that these values are quite similar to those obtained using the two different sets of user judgements with an average $\tau_{A/B} = 0.924$ with a quite low standard deviation $\sigma_{A/B} = 0.039$. For the BM25 and SVM retrieval models, the obtained average values are: $\tau_{BM25} = 0.797$, $\sigma_{BM25} = 0.090$ and $\tau_{VSM} = 0.754$, $\sigma_{VSM} = 0.125$, respectively.

Table 5.3: Kendall τ correlations of the personalization techniques for each of the 12 combinations of the user profiles configuration parameters: number of expanded terms, k, and normalization factor, p_0.

k	**5**			**10**			**20**			**40**		
p_0	**0.33**	**0.66**	**0.99**	**0.33**	**0.66**	**0.99**	**0.33**	**0.66**	**0.99**	**0.33**	**0.66**	**0.99**
$\tau_{A/B}$	0.947	0.896	0.896	0.974	0.844	0.974	0.922	0.922	0.974	0.896	0.922	0.922
τ_{XML}	0.947	0.844	0.922	0.922	0.974	0.896	0.740	0.844	0.922	0.922	0.870	0.948
σ_{us}	0.104	0.105	0.109	0.112	0.121	0.128	0.119	0.133	0.143	0.135	0.159	0.168
σ_{ASP}	0.145	0.150	0.152	0.155	0.158	0.163	0.160	0.170	0.178	0.171	0.190	0.199
τ_{BM25}	0.878	0.619	0.810	0.878	0.810	0.714	0.714	0.714	0.905	0.810	0.905	0.810
σ_{us}	0.053	0.038	0.032	0.063	0.054	0.058	0.072	0.067	0.072	0.086	0.093	0.102
σ_{ASP}	0.077	0.064	0.055	0.080	0.078	0.082	0.105	0.110	0.122	0.130	0.147	0.159
τ_{VSM}	0.619	0.524	0.714	0.810	1.000	0.810	0.714	0.714	0.714	0.905	0.714	0.810
σ_{us}	0.053	0.041	0.036	0.070	0.072	0.083	0.076	0.087	0.098	0.086	0.102	0.109
σ_{ASP}	0.071	0.055	0.057	0.093	0.101	0.112	0.113	0.130	0.142	0.129	0.153	0.165

These rather high values show that the ASPIRE and the user study rankings are very similar, independently of the user profile configuration being considered. This is particularly true when there are significant differences among the different values used to obtain the rankings (τ increases with higher values of σ_{us}). This fact has some sense, since in the case in which we obtain lower σ_{us} it might be difficult to be sure about the obtained ranking: the differences in performance among the different methods are rather small, thus making it more difficult (but also less important) to discriminate between them. Therefore, we can be pretty sure that ASPIRE is a reliable method to discriminate between (better and worse) personalization methods.

Selecting the best user profile configurations for a fixed personalization technique

As we want to select the best user profile configurations, now we will also calculate the rank correlations between the real and simulated results obtained by the different user profile configurations, for each of the 13 personalization techniques in the case of XML-based retrieval and the 7 personalization techniques that could be applied in the case of flat retrieval. These correlations are shown in Table 5.4, where we also show the standard deviation among the different values used to compute the rankings.

Table 5.4: Kendall τ correlations of the user profile configurations for each of the 13 personalization techniques.

	QE	NQE	HRR	SRR	IRR	I-HRR	p-HRR	CAS	CAS-or	NQE+m	HRR+m	SRR+m	IRR+m
$\tau_{A/B}$	1.000	0.939	0.879	0.788	0.818	0.848	0.931	0.727	0.424	0.667	0.879	0.939	0.939
τ_{XML}	1.000	0.970	1.000	0.909	0.879	0.818	0.473	0.576	0.545	0.939	0.939	0.909	0.909
σ_{us}	0.105	0.112	0.082	0.042	0.054	0.005	0.006	0.011	0.009	0.039	0.053	0.041	0.041
σ_{ASP}	0.113	0.112	0.086	0.048	0.061	0.012	0.007	0.018	0.013	0.034	0.042	0.032	0.032
τ_{BM25}	1.000	0.758	0.424	0.667	0.718	0.909	0.667	-	-	-	-	-	-
σ_{us}	0.087	0.079	0.016	0.015	0.016	0.040	0.007	-	-	-	-	-	-
σ_{ASP}	0.121	0.101	0.019	0.031	0.029	0.065	0.011	-	-	-	-	-	-
τ_{VSM}	1.000	0.909	0.545	0.697	0.121	0.848	0.333	-	-	-	-	-	-
σ_{us}	0.069	0.086	0.012	0.017	0.009	0.033	0.007	-	-	-	-	-	-
σ_{ASP}	0.098	0.110	0.016	0.019	0.017	0.051	0.006	-	-	-	-	-	-

Focusing on the XML-based model, the averaged Kendall $\tau_{XML} = 0.836$ is also high, with a standard deviation $\sigma_{XML} = 0.181$ higher than in the previous case. This fact is due to the existence of three personalization techniques where the correlations are not so high (p-HRR, CAS and CAS-or, see Table 5.4). As before, the performance of the different profile configurations under these methods is quite stable (they exhibit a very low value of standard deviation). In the same way, similar values has been obtained using different sets of human judgements, with a $\tau_{A/B} = 0.829$ and a standard deviation $\sigma_{A/B} = 0.154$.

With respect to the BM25 and VSM retrieval models, we obtain averaged values of $\tau_{BM25} = 0.735$, $\sigma_{BM25} = 0.186$, and $\tau_{VSM} = 0.636$, $\sigma_{VSM} = 0.322$, respectively. The explanation is the same, ASPIRE is able to rank with guarantee the different profile configurations only in those situations where the impact is relevant.

Therefore, it seems that when the differences in performance between the different user profile configurations of a given personalization method are important (higher standard deviations), ASPIRE is able to discriminate among them. When these differences are small (lower standard deviations), then it is not so critical to accurately distinguish which are the best user profile configurations.

An alternative to tackle these situations where is difficult to rank among the different approaches could be to obtain more data by increasing the number and types of queries. Although this approach would be expensive when using real users, it is not the case when using ASPIRE.

5.5.2 User study-ASPIRE Sieg et al. approach results comparison

To conclude the experimentation we would like to show the performance of ASPIRE when using the relevance criteria guidelines proposed by Sieg et al. [115] which is, as pointed out in Section 5.3, the closest approach to our proposal. In their work, the relevance criteria is to consider a document as relevant if it is classified under the ODP ontology concept being simulated, and not relevant otherwise. Our relevance criteria is similar, considering a document as relevant if it belongs to the area(s) of interest the given user profile represents, but not only that, but also if it has been retrieved by the IRS among the first *topkRel* results.

We have relaxed this last relevance criteria restriction (the main difference with Sieg et al. criteria) to see if it is really important or not. In order to perform this comparison we will use as ground truth the results obtained with the user study and Garnata. To simulate the lack of this restriction we have established $topkRel = 1500$, which is the Garnata maximum number of retrieved results for a given query. This approach will be denoted as *ASPIRE_S*, to emphasize that we are following the Sieg et al. relevance criteria guidelines.

In this experiment we will follow the same steps than in Section 5.5.1, i.e., we first perform a comparison between relevance assessments and then we evaluate the retrieval performance. As we will see, ASPIRE_S includes a big amount of noise into the relevance assessments, being therefore hard to obtain accurate retrieval results.

Relevance assessments comparison

As might be expected, the number of ASPIRE_S raw relevance assessments has grown a lot with the change from $topkRel = 100$ to $topkRel = 1500$, since *topkRel* limits the number of potentially relevant results. The user study number of relevance assessments is obviously still 2362, while the number of current ASPIRE_S relevance assessments is 17373 (before it was 1965). This change introduces a large number of false positives in the ASPIRE_S assessments (around 85% on average) and also, the number of relevance assessments are softly correlated exhibiting a Pearson correlation value of 0.285.

In order to measure the new automatic assessments we will consider the average results across all the evaluation triplets, of precision, recall and F metrics, together

Table 5.5: Averages and standard deviations of precision, recall and F metrics across the 126 evaluation triplets, with $topkRel = 1500$.

	pre	**rec**	**F**
μ	0.190	0.753	0.221
σ	0.276	0.295	0.245

with the standard deviations, displayed in Table 5.5. We can still observe large deviations (quite different behaviour depending on the query and the profile being evaluated) together with a notable deterioration of precision and F measures. Thus, only around 20% of the simulated relevance assessments are correct. However, we can observe a much higher recall, which is reasonable, since we let much more relevance assessments come into play. We will focus on the performance comparison between the results using the user study and ASPIRE Sieg-based relevance criteria, which in this case seems more difficult considering the previous overlap degree and number of false positives.

Retrieval performance evaluation results comparison

This section also pursues the same two objectives as the section with the same name under our proposed evaluation framework, i.e., 1) to test whether ASPIRE_S should be considered as a reliable approach in the evaluation of personalized IRSs, and 2) to show whether ASPIRE_S is able to rank properly the personalization approaches and the different profile representations.

Is ASPIRE_S a reliable evaluation approach?

Firstly, we consider as independent each of the 149 personalization technique and profile parameter configurations. For each configuration we focus on the quality of the rankings obtained for each evaluation triplet, trying to measure how the 126 ASPIRE_S-based NDCGs correlates with the NDCGs obtained using the real user assessments.

Summarizing, the Pearson correlation values vary in the range $[-0.051, 0.655]$, with an average correlation equal to 0.325 and a relatively large standard deviation of 0.185. These correlation coefficients are plotted in Figure 5.6 (in the y axis) against the averaged NDCG values from the user study (in the x axis). In this figure we can see how the results of the performance comparison are almost randomly distributed

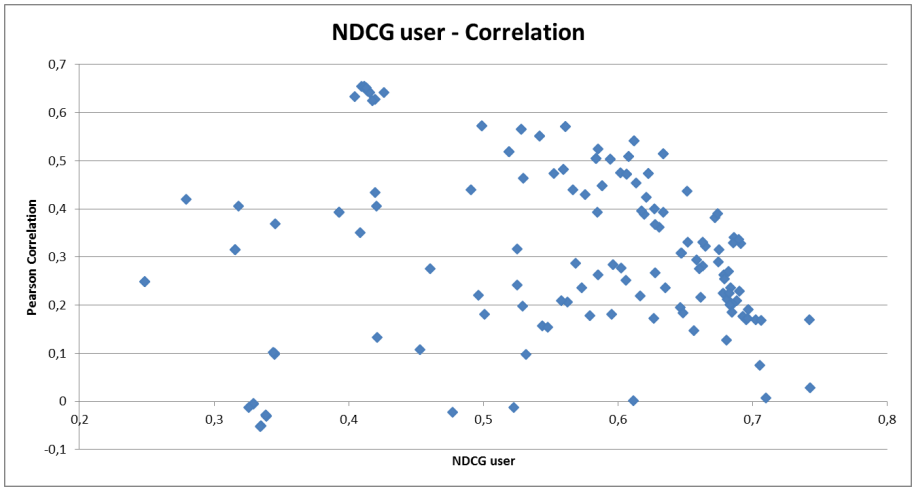

Figure 5.6: NDCG user study-ASPIRE_S correlations (y axis) against the averaged NDCG values from the real study (x axis), with $topkRel = 1500$.

in the range $[-0.051, 0.655]$ (compare this figure with Figure 5.3).

Once we have evaluated the intra-configuration comparison, now we will focus on how these configurations relate to each other. In this case, we consider an evaluation matrix with 149 cells, but now each of these cells represents the average NDCG obtained after running the 126 evaluation triplets under a given configuration.

Figure 5.7 plots the user study averaged NDCG values (x axis) against the corresponding values obtained from ASPIRE_S (y axis), for the 149 evaluation matrix cells (compare it with Figure 5.4). In this case, none of the conclusions drawn from our proposed ASPIRE framework are fulfilled, since the current ASPIRE_S results are not close to the user study ones. The linear regression labelled as *LR (user/ASPIRE_S)* shows an R-squared value of 0.0092 (0.096 Pearson correlation coefficient), i.e., both variables are almost independent. We also can see that there is a big number of results on both sides of the ideal fit line, labelled as *LR ($y = x$)*, sometimes overestimating and sometimes underestimating the real different personalization techniques performance.

Stability of rankings of ASPIRE_S-user study personalization techniques and user profile configurations using the new relevance criteria

The next step is to test whether we may trust on the systems ranking provided

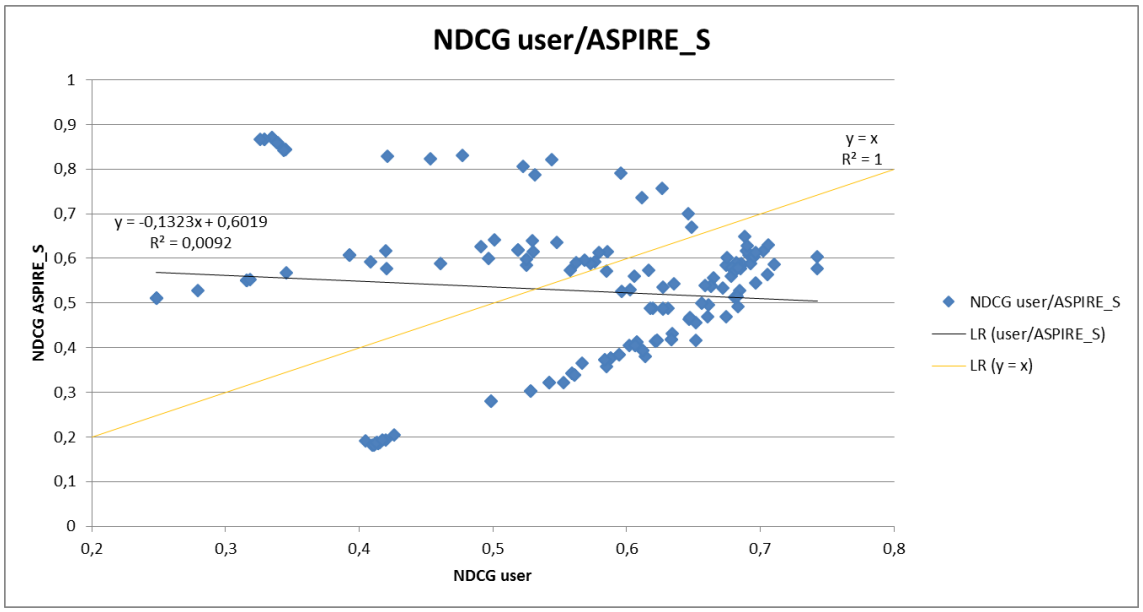

Figure 5.7: Averaged NDCG values from ASPIRE_S (y axis) against the averaged NDCG values from the real study (x axis), for each one of the 149 personalization techniques-user profile configuration parameters combinations, with $topkRel = 1500$.

Table 5.6: ASPIRE_S: Kendall τ correlations of the personalization techniques for each one of the 12 combinations of the user profiles configuration parameters: number of expanded terms, k, and normalization factor, p_0.

k	**5**	**5**	**5**	**10**	**10**	**10**	**20**	**20**	**20**	**40**	**40**	**40**
p_0	**0.33**	**0.66**	**0.99**	**0.33**	**0.66**	**0.99**	**0.33**	**0.66**	**0.99**	**0.33**	**0.66**	**0.99**
τ_{XML}	0.447	0.221	0.013	0.247	0.013	-0.091	0.091	-0.143	-0.013	-0.091	-0.065	0.039
σ_{ASP_S}	0.188	0.192	0.194	0.192	0.196	0.197	0.191	0.193	0.193	0.192	0.191	0.188

by ASPIRE_S. We use Kendall τ rank correlation to look at the stability of rankings.

Selecting the best personalization techniques for a fixed profile

We are going to compute the rank correlations between the real and simulated personalization techniques results, for each of the 12 user profile configurations. These correlations are displayed in Table 5.6. We also include the standard deviation of the NDCG over the different personalization techniques using ASPIRE_S. An averaged Kendall $\tau_{XML} = 0.056$ with a large standard deviation $\sigma_{XML} = 0.172$ is obtained from the 12 user profile configurations.

These low values and large deviations show that the ASPIRE_S and the user study rankings are very different, independently of the user profile configuration being considered. Therefore, we can not ensure that ASPIRE_S is a reliable method to discriminate between (better and worse) personalization methods.

Selecting the best user profile configurations for a fixed personalization method

We are going to compute the rank correlations between the real and simulated different user profile configurations results, for each of the 13 personalization techniques. These correlations are shown in Table 5.7. In this case the averaged Kendall $\tau_{XML} = 0.329$ is also low, and the standard deviation $\sigma_{XML} = 0.699$ very high. This low average value and large deviation show that there are big differences between the different personalization techniques. Although the σ_{ASP_S} values are low, we can not ensure that ASPIRE_S is a reliable method to discriminate between (better and worse) user profile configurations.

To conclude this section, and considering all the results obtained during the comparison of ASPIRE_S with the results obtained using a user study (which considers

Table 5.7: ASPIRE_S: Kendall τ correlations of the user profiles for each one of the 13 personalization techniques.

	QE	**NQE**	**HRR**	**SRR**	**IRR**	**I-HRR**	**p-HRR**	**CAS**	**CAS-or**	**NQE+m**	**HRR+m**	**SRR+m**	**IRR+m**
τ_{XML}	1.000	0.121	-0.818	-0.636	-0.576	0.818	-0.424	0.545	0.606	0.788	0.970	0.939	0.939
σ_{ASP_S}	0.046	0.040	0.082	0.073	0.059	0.006	0.011	0.030	0.028	0.062	0.077	0.045	0.045

the real user interactions with the system), we can state that the relevance criteria guidelines proposed by Sieg et al. seems to be not useful for a personalized retrieval system automatic evaluation.

5.6 Conclusions and future work

In this chapter we have faced the difficult problem of personalized IRSs evaluation. Without any doubt, the inclusion of personalization is every day more and more frequent in a broad variety of services. This tendency shows the importance of being able to build efficient and robust personalization techniques to be part of these services. The evaluation step of any personalized system is a crucial stage in their development and improvement. Indeed, high efforts are made to evaluate personalized systems. We have reviewed several methodologies for the evaluation of personalized IRSs in the literature, but all of them have some disadvantages in one or another way.

Considering the previous facts, we have proposed an automatic evaluation methodology for personalized IRSs. This methodology joins the advantages of the system-centred and user-centred evaluation approaches producing repeatable, comparable and generalizable results together with the inclusion of the user context within the evaluation process. We must specify that the proposed evaluation approach is focused on maximizing the retrieval effectiveness, leaving aside the evaluation of the user-IRS interactions. The only requirements to use ASPIRE is to have a document collection where its documents (at least part of them) are able to be classified into different categories, and a suitable set of queries for this collection.

Moreover, we have validated ASPIRE by comparing its results with those obtained from a carried out user study. Not many evaluation approaches present this validation process which, in our opinion, is a key factor to trust on the proposed evaluation methodology.

Some ASPIRE reliability metrics have been proposed regarding both, the automatically generated relevance assessments and the evaluation performance of the retrieved results. Although the simulated relevance assessments are not completely similar to those obtained from the user study (around 75% of the real relevance assessments are compatible with the basic assumption used to generate the simulated relevance assessments, and there is an overlap degree of around 50% between the real and simulated relevance assessments), they are good enough to get very similar evaluation results. Figure 5.4 is very clarifying, showing Pearson correlation values higher than 0.914 between both sets of results.

We have also shown how ASPIRE may be used to select the best personalization techniques from a set of them, or the best user profile configurations for a given personalization technique. The high correlation values between the rankings obtained for different personalization strategies and different profile configurations, by ASPIRE and the user study, give an idea of the expected reliability of these selections.

We should also mention that in our evaluation tests we have used a very heterogeneous set of personalization techniques, ranging from very good to very bad performance ones, obtaining good results in all cases. In addition, ASPIRE has been tested with Garnata, BM25 and Vector Space retrieval models showing similar results, thus reinforcing the fact that ASPIRE is robust and, at the same time, independent on the type of the used collection (XML or flat), being applicable in any of these circumstances. This fact demonstrates that ASPIRE is not only a reliable evaluation approach but also robust. In this line, we have also carried out comparisons with a state-of-the-art approach, very similar to our proposal. In this case, we have shown we have to be very cautious with the results derived following the relevance criteria guidelines proposed by Sieg et al. [115].

On the other hand, it should be clear that ASPIRE does not pretend to completely replace user studies, since it is very important to collect qualitative information about the IRS from real users. It rather should be considered as an easy, fast and reliable alternative to them. To have a reliable and robust evaluation methodology is a very good resource, specially indicated for the first stages in the development of personalization techniques, or when a user study is not possible due to any circumstances, such as the lack of resources or time, or for example to pre-analyse the

expected performance for those queries that should be used in a real user study. ASPIRE also helps to make final user studies experimentation more worthwhile, by limiting the number of personalized IRS configurations users should evaluate.

As future work, we would like to explore some other criteria for the automatic relevance assessments generation and to extend ASPIRE to also incorporate the user-IRS interactions into the automatic evaluation process.

Chapter 6

User Profiles

6.1 Introduction

For a given query, IRSs try to retrieve the most accurate but, at the same time, general and diversified results in order to please all possible users. This previous fact, together with the incredible huge amount of available information nowadays, make IRSs to have difficulties to provide the right answers, in order to satisfy the specific information needs of a given user. To avoid this situation, modern IRSs are moving from a system-centred to a user-centred behaviour, incorporating personalization techniques to adapt their results to specific users.

Personalization techniques need to use the user information, which will be stored in the user profile. A user profile is a representation of the user context, which may include interests, preferences, personal data, location, etc. The main objective of any personalization process is how to best represent and exploit this user information, with the intention to fit the results to users the best as possible. Thus, these personalized results will hopefully better and faster cover the user information needs, making this user to be more satisfied with the IRS.

The user profile building process is quite difficult, since user interests and preferences are not easy to be captured and they also change over time [70, 95]. However, it is a very important step in order to obtain good personalized results, which will highly depend on the user profile quality and how well it is exploited in the retrieval process. There are three main steps in a user profile building process: 1) how to

gather the user information, 2) how to represent this information, and 3) how to keep this information updated. All these steps are well explained in Section 2.2.

Owing to the importance of having a good user profile representation, this chapter focuses on the analysis of different ways to build user profiles. Concretely, we shall focus on *simulated content-based user profiles*, and more specifically on the user interests and preferences. These profiles are frequently used in contextual evaluation environments, such as [115]. Moreover, this kind of profiles may be ideal for the introduction of personalization in privacy-constrained environments, in which users are reluctant to reveal their personal information. These environments are very frequently becoming a significant barrier for the personalized IRSs common use [114]. We have concretely faced this problem with the Andalusian Parliament, where the members of the parliament do not allow any personal data collection of themselves nor the citizens. In this way, any IRS could integrate personalization techniques to improve its retrieval performance and user satisfaction, by only giving the user the possibility to choose with which of the simulated profiles he/she is more alike.

The only requisite to build simulated content-based user profiles is that the document collection documents, or at least a subset of them, must be able to be classified into different areas of interest or categories, in which future users could be interested. In our particular case, we use the records of the AP proceedings, and more concretely the *committee sessions* (see Section 4.4.1), which are devoted to different areas of interest or categories, such as agriculture, education or economy. Each of these records (or documents) contains the full transcriptions of the members of the parliament speeches in each parliamentary session, where laws are passed or different issues of interest are discussed. The main component of these documents is the *initiative*, with an average number of 5.6 initiatives per document, which presents a detailed discussion about a specific issue. Each of these initiatives is tagged with one or more subjects extracted from the EUROVOC thesaurus, being manually assigned by parliamentary documentalists as the best representation of its content. Since each of these documents belongs to one committee, we have an implicitly classified document collection. If this were not the case, a clustering process could be used to find clusters of similar documents according to their content, and later a classification process may assign new documents to the corresponding clusters.

While these content-based simulated user profiles could be considered as *a lacking of 'reality'*, since they do not represent real users, they are a valid approach [33, 115] for possible users interested in some areas of interest. In this situation, the user might also choose to include several terms in the query describing the committee content, terms that could be difficult to select for a normal user, also appearing the query-drift problem. On the other hand, the user might also choose to filter out the documents which do not belong to the committee, but in this case, there might be relevant results which are not shown to the user (around 25% in our studies).

If we join the recent rise of personalized systems, together with the fact that their evaluation through user studies is rather complicated (due to the large required resources such as, access to real users, time, money or even the needed infrastructure for their implementation), we consider particularly important to test and improve the quality of content-based user profiles.

As a first approach, we have developed a user profile only based on terms (independently of where they appear in the document). Secondly, we have built a user profile based on the EUROVOC thesaurus subjects, manually assigned to each initiative discussed in a committee session, and thirdly, we have configured a hybrid user profile composed of both, terms and subjects. Finally, we present the way to properly use them, especially the hybrid approach, and a comparative study between the final six different user profile representation approaches. In this evaluation, we have obtained quite good personalized results in terms of retrieval performance, and some interesting conclusions about the goodnesses of these content-based approaches.

The rest of the chapter is organized as follows. Section 6.2 shows the developed user profiles and how they are built and used. Section 6.3 shows the experimental design, the evaluation and the obtained results from the previous user profile approaches. Finally, Section 6.4 shows the conclusions and proposals for further research.

6.2 Developed user profiles

The quality of personalized results will highly depend on the user profile quality and how it is exploited in the retrieval process. Hence, the user profile building process is one of the most important steps to obtain good personalized results. From the three

main steps in the user profile building process, i.e., user information gathering, user profile representation and user profile update, we will focus on the second stage, since this chapter main goal is to make a comparative study between different user profile representations performance, in order to find the best representation approach.

Due to the frequent important restriction concerning to the collection of user personal information, and additionally to the difficulty to have accurate and updated user profiles, we have decided to build simulated user profiles based on content, concretely on the transcriptions of the AP working committees, where much of the work of the parliament takes place. Thus, assuming that the citizens interests and preferences might be represented by the topics in a given committee, we analyse its content to learn the corresponding profile. Content-based user profiles have other advantages, such as, the user information gathering step is not necessary and to keep the user profile updated is enough to simply update it with each document collection documents change.

We have developed and carried out a comparative study between six different user profile representation approaches, considering the document terms, the initiative manually assigned subjects, and hybrid approaches considering both at the same time. The obtained performance results from this comparative study are pretty good. We also present some interesting conclusions together with some advantages of these content-based user profiles over user profiles from real users.

As the reader may recall, we have already built and used these simulated content-based user profiles in the previous chapters, see Section 4.4.2. These user profiles were only based on terms, selected from the AP committee documents. As we mention in the previous section, at this point of our research, our main objective was to measure the retrieval performance of different personalization techniques, more than build the best possible user profiles. Consequently, we took a simple and common approximation to build the user profiles, selecting a weighted keyword-based representation and following the well-known *tf*idf* approach. Concretely, each profile associated to an area of interest (committee) was comprised by the first k terms of this area documents, ordered by decreasing *tf*idf* and weighted by *idf*.

We chose to weight the user profile terms by *idf* because each term is better represented by this value than by the *tf*idf* value, considering the full corpus. But, this affirmation is no longer true when we consider subjects instead of only terms.

Subjects are actually some kind of metadata-tags of the initiative content, usually from a controlled vocabulary (in our case from the EUROVOC thesaurus). Since this is a very synthesized information, even in our case being manually assigned by expert documentalists, the concept of *idf* (diminish the influence of the corpus frequent terms) makes no sense for subjects. Moreover, the resulting content-based *tf*idf* user profiles from the previous chapters included some terms not actually important or representative of any area of interest, which are more related to the documents format, such as, *señor(sir)* to introduce a new speaker or *gracias(thank you)* to express gratitude at the end of the speaker speech, between others. Since our objectives were others than build the best user profiles, we decided to manually delete this kind of terms. We present a comparison between the old way to build the user profiles based on terms and the new approach in Section 6.3.

With the intention to avoid the two previous problems, i.e., to have a homogeneous way to build content-based user profiles independently of the content we are considering, and to do not have to manually delete the previous kind of terms, more related to the documents format than to be representative of the profile itself, we propose the next natural, easy, but at the same time effective way to build our new content-based user profiles.

6.2.1 Building process

First of all, we next explain the characteristics and components for each of the three proposed user profiles based on terms, subjects and both at the same time (see examples in Table 6.1):

- **tProf**: The first profile approach, based on the collection terms, can be considered as a *weighted keyword* profile, since the terms themselves are the items which represent the user interests. These profiles are the easiest to build, but they need to have many terms to accurately define a user interest. These profiles are also less understandable for users than those based on concepts, since their interests are much easily mapped with concepts than with isolated terms. But at the same time, terms allow a more fine-grained representation of the collection content.

Table 6.1: Examples of the three proposed user profiles, for the 'agriculture and livestock' area of interest (unstemmed and translated into English).

tProf	$t =$ { 0.007*agriculture 0.007*sector 0.004*fishing 0.004*agrarian 0.004*production 0.003*aid 0.003*farmer 0.002*product 0.002*rural 0.001*oil ... }
sProf	$s =$ { *0.216*"agriculture aid" 0.127*"agricultural policy" 0.098*"agricultural production" 0.098*"oily" 0.095*"food industry" 0.091*"fishing" 0.083*"oil" 0.075*"huelva province" ...* }
stProf	s_1 =*0.216*"agriculture aid"* t_{s1} = {0.007*aid 0.006*sector 0.006*agriculture 0.005*farmer ...} s_2 =*0.127*"agricultural policy"* t_{s2} = {0.009*agriculture 0.007*agrarian 0.006*production ... } ⋮ ⋮

- **sProf**: This second approach, based on the initiative subjects, can be considered as a *weighted concept* profile, since these subjects represent abstract topics of interest for the user instead of terms. They are represented as vectors of weighted concepts, without any structure. General concepts profiles main assets are their robustness to vocabulary variations and a less requirement of user feedback. These characteristics and the fact that the subjects are manually selected by experts in the document collection, as the best content representation for the parliamentary initiatives, made us to think they would be a good resource to exploit.

- **stProf**: The third profile approach, based on subjects and terms, is a hybrid approach among the weighted concept and weighted keyword profiles, keeping concept abstraction but enriched by the terms fine-grained contribution. To build this profile we learn the most representative terms for each collection subject. Thus, this new profile now contains two levels: the first, with the subjects which represent the profile, and the second composed by the terms which represent each first level subject.

We now show the way we select the elements of each type of profile. Let X represent either a term in the case of *tProf* or a subject in the case of *sProf*, and let Y represent a profile. Then we define $f^+(X,Y)$ as the frequency of X in documents belonging to any area(s) of interest which form the profile Y; $f^+(Y)$ is the number of elements (either terms for *tProf* or subjects for *sProf*) within Y; $f^-(X,Y)$ and $f^-(Y)$ are respectively the frequency of X and the number of elements in documents outside the profile Y. For the *stProf* profiles, X represents a term and Y represents a subject, $f^+(X,Y)$ being in this case the frequency of X within initiatives classified

Table 6.2: Final *tProf* and *sProf* user profiles using $exp[Terms|Subj] = 5$ and $p_0 = 0.66$.

sProf	*0.66*"agriculture aid" 0.388*"agricultural policy" 0.299*"agricultural production" 0.299*"oily" 0.290*"food industry"*
tProf	`0.66*agriculture 0.647*sector 0.401*fishing 0.399*agrarian 0.398*production`

by the subject Y and $f^+(Y)$ the total number of terms within these initiatives; $f^-(X,Y)$ and $f^-(Y)$ have in this case the obvious meaning. We then define the relevance of X with respect to Y, $R(X,Y)$ as:

$$R(X,Y) = \frac{f^+(X,Y)}{f^+(Y)} - \frac{f^-(X,Y)}{f^-(Y)} = \begin{cases} \leq 0 & \text{delete} \\ > 0 & \text{sort in } \downarrow \text{ order} \end{cases} \tag{6.1}$$

that is, the normalized frequency of X within Y minus the normalized frequency of X outside Y. If the final value is $R(X,Y) \leq 0$, it means that X is more frequent outside than within Y, so it is not representative of Y and we will not consider it. However, if the final value is $R(X,Y) > 0$, this means that X represents Y at a certain degree, so we keep it. All the retained elements are sorted in decreasing order of relevance to form the final user profile. In the case of the *stProf* profile, we first calculate the list of subjects and next the list of terms associated to each subject.

Once we have defined the three different content-based user profiles, we are going to show how we have used them in our evaluation process and how we have solved a small problem, which appears when we try to use the *stProf* profiles, which has led us to propose four different variations of the original *stProf* approach.

1) *tProf and sProf*: The use of term-based and subject-based user profiles is quite simple. It basically involves taking the top-k relevant terms (*expTerms*) or subjects (*expSubj*). Once we have these first k *expTerms* or *expSubj*, we normalize (proportionally) their weights with a maximum normalization value p_0. The combination of k *expTerms* or *expSubj* with p_0 gives us a total number of $k * p_0$ different weighted term or subject sets, to provide to each personalization technique. Check Table 6.2 to see an example of these final user profiles from Table 6.1.

2) *stProf*: Its use is somewhat more complicated. In principle, the process should be to get the first *expSubj* profile subjects, and for each of these subjects to get the given first *expTerms* terms. Each term weight will be multiplied by its corresponding

subject weight. Thus the terms, which will be the ones finally used by the personalization techniques, will already incorporate in their weights the influence of their subjects.

But we find a problem in the previous process: when joining the different terms associated to different subjects, some of these terms are repeated (several subjects have terms in common, as *agriculture* in the example of Table 6.1). Since having repeated terms with different weights makes no sense, we consider the following approaches to fix the weights of these terms:

a) **stProf_add** *(add weights)*: collapse the repeated terms into one, with a weight equal to the addition of the individual weights.

b) **stProf_max** *(maximum between weights)*: we only keep the repeated term with the highest weight, removing all the other repeated terms.

c) **stProf_addFill** *(add weights, filling terms)*: same as *stProf_add*, but each time a term is deleted from a subject, the next one in the list is included until having *expTerms* terms for each subject.

d) **stProf_maxFill** *(maximum between weights, filling terms)*: same as *stProf_addFill*, but using the maximum instead of the sum.

The first two approaches involve that we do not always obtain the same number of terms for the personalization techniques, as it happens with the last two approaches. It should be noted that, in the last two approaches the filling process should start from the last *expSubj* subject, since we want more information from the most profile representative subject, i.e., the first *expSubj* subject. At the end of this process, the final terms will be also normalized with a maximum normalization value p_0. The combination of the *expTerms*, *expSubj* and p_0 gives us a total number of $expTerms * expSubj * p_0$ different weighted term sets to provide to each personalization technique.

We can see examples of the previous four different *stProf_** user profile approaches looking at Table 6.3, based on the *stProf* user profile from Table 6.1. Some of their characteristics are: 1) all approaches first term has a weight equal

Table 6.3: Final *stProf* user profile using $expSubj = 2$, $expTerms = 3$ (to make it more clear and shorter), and $p_0 = 0.66$.

stProf_add	0.66*agriculture 0.421*aid 0.372*sector 0.237*agrarian 0.219*production
stProf_max	0.66*aid 0.583*sector 0.548*agriculture 0.371*agrarian 0.344*production
stProf_addFill	0.66*agriculture 0.421*aid 0.372*sector 0.307*farmer 0.237*agrarian 0.219*production
stProf_maxFill	0.66*aid 0.583*sector 0.548*agriculture 0.482*farmer 0.371*agrarian 0.344*production

to 0.66 (p_0); 2) while the first two approaches have five terms instead of six, because the 'agriculture' term is repeated in both *expSubj*, the last two approaches have six terms, since they fill terms until $expSubj * expTerms$; 3) since the term 'agriculture' is repeated, the add of their weights places it as the first term in the **add** approaches, while 'aid' is the first term in the **max** approaches; 4) the term 'farmer', which belongs to the "agriculture aid" subject, is the filled term by the **Fill** approaches (remember that the filling process starts from the last *expSubj* subject).

6.3 Experimental evaluation and results

This section shows the corresponding results for the different proposed ways to build user profiles based on terms, subjects and the four approaches using both at the same time, together with the derived conclusions from each approach.

The evaluation framework is composed by the components exposed in Section 4.4. Summarizing, we have used Garnata as the search engine, the heterogeneous set of 23 queries formulated by real users of the document collection, the relevance assessments were obtained from the carried out user study, which involved 31 users, with a total number of 126 evaluation triplets (user, query, profile). NDCG has been used as the evaluation metric, with the special considerations shown in Section 4.5.1 because of the structured nature of the documents. The used personalization techniques are *NQE, HRR, SRR, IRR, NQE+m, HRR+m, SRR+m, IRR+m, CAS and CAS-or*, which represent a highly heterogeneous set of personalization techniques. The only different or slightly modified components are the following: on the one hand, the document collection is exactly the same with the only difference of the inclusion of each initiative corresponding subjects, on the other hand, we have de-

Table 6.4: Maximum, average (μ) and std. (σ) performance values for the *tf*idf* and the *current* approach user profiles.

		NQE	HRR	SRR	IRR	NQE+m	HRR+m	SRR+m	IRR+m	CAS	CAS-or
$tf * idf$	max	0.621	0.627	0.601	0.595	0.651	0.667	0.572	0.572	0.627	0.627
	μ	0.482	0.544	0.560	0.550	0.609	0.617	0.530	0.530	0.602	0.586
	σ	0.095	0.061	0.030	0.039	0.041	0.049	0.036	0.036	0.019	0.041
current	max	0.634	0.652	0.625	0.620	0.678	0.696	0.597	0.597	0.675	0.668
	μ	0.536	0.593	0.596	0.589	0.624	0.630	0.540	0.540	0.656	0.645
	σ	0.078	0.047	0.020	0.025	0.051	0.057	0.040	0.040	0.014	0.014

veloped an additional personalization technique called *CAS-mix*, which is only used with the user profiles based on subjects.

Hereafter, in this section tables of results, if not otherwise specified, each cell represents the average over the 126 evaluation triplets from the carried out user study, for a given combination of expansion terms or subjects, $k = 5, 10, 20, 40$ (for *stProf* profiles, k subjects and $l = 1, 5, 10$ expansion terms for each k subject), maximum normalization factor $p_0 = 0.33, 0.66, 0.99$, and a given personalization technique.

6.3.1 Profiles based on terms

We start showing the results from the user profiles based on terms. First of all, we want to illustrate the performance differences between the results obtained following the old way to build these user profiles based on *tf*idf* and the new way to build them, exposed in Section 6.2 and labelled as *tProf*. Table 6.4 shows the best, average and standard deviation performances of the *tf*idf* and our current approach for the user profiles based on terms, under the above evaluation framework.

As we can see in this table, the maximum and average NDCG values are always higher (better) under our current approach for building the user profiles. Moreover, for all, except the *+m* personalization techniques, the standard deviations are also lower, which indicates more homogeneous results across the different user profile configuration parameters k and p_0.

Thus, the way to build user profiles based on content exposed in this chapter is a better approach than those profiles based on the previous *tf*idf* version, since it

Table 6.5: NDCG averaged values for the user profiles based on terms.

k	p_0	**NQE**	**HRR**	**SRR**	**IRR**	**NQE+m**	**HRR+m**	**SRR+m**	**IRR+m**	**CAS**	**CAS-or**
5	0.33	**0.634**	0.637	0.576	0.576	0.526	0.526	0.472	0.472	0.638	0.645
5	0.66	0.593	0.621	0.606	0.605	0.612	0.613	0.528	0.528	0.661	0.667
5	0.99	0.546	0.599	0.609	0.604	0.650	0.660	0.554	0.554	0.668	**0.668**
10	0.33	0.632	**0.652**	0.601	0.601	0.548	0.550	0.487	0.487	0.631	0.641
10	0.66	0.559	0.614	**0.625**	**0.620**	0.637	0.639	0.546	0.546	0.658	0.653
10	0.99	0.491	0.576	0.604	0.595	0.666	0.674	0.572	0.572	0.662	0.656
20	0.33	0.603	0.636	0.605	0.605	0.581	0.578	0.501	0.501	0.641	0.632
20	0.66	0.509	0.584	0.609	0.600	0.656	0.663	0.560	0.560	0.661	0.644
20	0.99	0.446	0.539	0.585	0.569	0.671	0.690	0.586	0.586	0.665	0.648
40	0.33	0.568	0.614	0.604	0.601	0.599	0.598	0.511	0.511	0.646	0.624
40	0.66	0.457	0.547	0.574	0.561	0.668	0.677	0.567	0.567	0.669	0.632
40	0.99	0.389	0.494	0.553	0.533	**0.678**	**0.696**	**0.597**	**0.597**	**0.675**	0.634
μ		0.536	0.593	0.596	0.589	0.624	0.630	0.540	0.540	0.656	0.645
σ		0.078	0.047	0.020	0.025	0.051	0.057	0.040	0.040	0.014	0.014
Baseline		0.388									

is independent of the kind of content considered in the building process and, at the same time, has demonstrated to obtain a better performance.

Having demonstrated the suitability of the new way to build content-based user profiles, Table 6.5 shows the NDCG average values considering the user profiles based on terms for all profile configuration parameters under the above evaluation framework. From this table we may draw the following main conclusions: 1) personalized results, independently of the user profile configuration parameters, are always better than the original not personalized result (baseline); 2) the best k and *p_0* user profile parameter combination for each personalization technique (in boldface) depends on this given personalization technique, being the highest values for those techniques which best avoid the *query-drift* problem (except for *CAS-or*), and relatively low values for those techniques which partially avoid this problem; 3) the absolute and averaged maximum performances are respectively obtained by *HRR+m* and *CAS*, with $k = 40$ and $p_0 = 0.99$ in the maximum case. The absolute maximum performance represents an improvement of 79.38% over the baseline.

Table 6.6: NDCG averaged values for the user profiles based on subjects.

k	p_0	NQE	HRR	SRR	IRR	NQE+m	HRR+m	SRR+m	IRR+m	CAS	CAS-or	CAS-mix
5	0.33	0.579	0.594	0.537	0.537	0.488	0.488	0.445	0.445	0.543	0.551	0.657
5	0.66	0.533	0.583	0.569	0.564	0.557	0.559	0.491	0.491	0.540	0.551	0.663
5	0.99	0.481	0.552	0.564	0.553	0.588	0.593	0.517	0.517	0.532	0.553	0.665
10	0.33	**0.588**	**0.603**	0.551	0.551	0.503	0.503	0.457	0.457	0.544	0.569	0.657
10	0.66	0.531	0.585	**0.577**	**0.572**	0.577	0.584	0.504	0.504	**0.552**	0.570	0.663
10	0.99	0.467	0.547	0.565	0.550	0.611	0.623	0.534	0.534	0.551	0.564	0.667
20	0.33	0.584	0.600	0.563	0.563	0.512	0.510	0.465	0.465	0.520	0.572	0.667
20	0.66	0.513	0.573	0.576	0.568	0.589	0.597	0.514	0.514	0.536	0.571	0.672
20	0.99	0.441	0.519	0.556	0.540	0.627	0.636	0.544	0.544	0.544	0.566	0.674
40	0.33	0.562	0.588	0.563	0.561	0.518	0.521	0.471	0.471	0.489	**0.578**	0.673
40	0.66	0.475	0.542	0.562	0.552	0.594	0.604	0.521	0.521	0.513	0.575	0.675
40	0.99	0.406	0.490	0.538	0.521	**0.632**	**0.645**	**0.552**	**0.552**	0.526	0.564	**0.679**
μ		0.513	0.565	0.560	0.553	0.566	0.572	0.501	0.501	0.532	0.565	0.668
σ		0.060	0.035	0.013	0.014	0.050	0.055	0.035	0.035	0.018	0.009	0.007
Baseline		0.388										

6.3.2 Profiles based on subjects

As it has been explained in its definition, *sProf* user profiles only use the initiative subjects in their build process. Just to remember, each document initiative may has one or several subjects, manually selected by human documentalists as the best representation of the initiative content, while each subject may be composed by one or several words. The results of applying these user profiles based on subjects under the given evaluation framework are presented in Table 6.6.

However, before to extract some conclusions from the previous table results, we are going to explain how subjects are actually used by the different evaluation framework personalization techniques. A priori, this use should be just like when using the user profiles based on terms (*tProf*), but because of the subjects own nature, this is not exactly accurate.

Subject words are used as the expansion terms under the *NQE, HRR, SRR, IRR, NQE+m, HRR+m, SRR+m* and *IRR+m* personalization techniques. We need to note two slight differences with respect to the *tProf* user profiles: the first difference is about the expansion process, i.e., although k subjects are still used in this expansion process, but as subjects are composed by several words (including some potentially repeated words between subjects), the total number of expansion words is unlikely to be equal to k. The second difference is that although the subjects words are obviously semantically related to its corresponding initiative content, the words

themselves do not necessarily match the initiative terms. Even though these two previous differences turn a result comparison between *tProf* and *sProf* profiles, by each k and p_0 user profile configuration parameters not totally comparable, the tendencies and general conclusions are still valid and very similar actually. In this sense, we can see how *sProf* profiles results, shown in Table 6.6, for the first eight personalization techniques are lower than those for *tProf* profiles in Table 6.5, but at the same time how the maximum values, averages and even standard deviation tendencies are quite similar.

The use of the *CAS* and *CAS-or* personalization techniques is also a bit different from the *tProf* user profiles approach, see Section 4.3.3. Just to remember, the *CAS* personalization technique underlying CAS query in the previous case was:

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

while in the *CAS-or* underlying CAS query, each *profileTerm* is within a different *about* clause, being all of them connected by *or* gates. Now, considering subjects instead of terms, the *CAS* and *CAS-or* underlying CAS queries are transformed into the following expressions, respectively:

```
//MaxUnit[about(.//materias,profileSubjects)]//*[about(.,originalQueryTerms)]
//MaxUnit[about(.//materias,profileSubject1) or about(.//materias,profileSubject2)
   or...or about(.//materias,profileSubjectK)]//*[about(.,originalQueryTerms)]
```

As it is obvious, in the new *CAS* queries using the *sProf* user profiles, the previous *profileTerms* are substituted by *profileSubjects*. But, as the reader may observe, these profile subjects are now only searched in the initiative associated *materias* tag, where subjects are located, and not in the whole *MaxUnit* content. Therefore, the new *CAS* queries for *sProf* profiles search the original query terms anywhere in the document, but only results with initiatives where *profileSubjects* appear will be actually retrieved (the higher the number of *profileSubjects* the better). *CAS-or* relaxes this last requisite, i.e., the number of required *profileSubjects*.

We have decided to allow *sProf* CAS approaches to search only in the initiative *materias* tags to avoid the previous *sProf NQE,HRR,...,IRR+m* observed unmatching problem between the initiative assigned subjects and its content terms (although both are semantically related).

Table 6.6 for *sProf* user profiles shows a considerably lower performance for *CAS* and *CAS-or* personalization techniques in comparison with their Table 6.5 *tProf* counterparts. As a consequence, we made substantial efforts to implement and test many design variations for both CAS approaches, including not to propagate to the *MaxUnit* but the *initiative* and other structural units without success, between others. We even tried the same *tProf* CAS approaches behaviour, i.e., not to search the subjects in the *materias* tags but anywhere in the content, which may be considered as the equivalent behaviour of *sProf* NQE, HRR,...,IRR+m approaches. Although the results slightly improved, this would imply to consider subjects as simple terms, and since content and structure queries allow to search in specific places, we want to treat subjects as subjects and not as terms.

Due to the previous CAS approaches low performance results, we have developed a new hybrid CAS personalization technique denoted as *CAS-mix*, which mixes subjects from the *sProf* profiles and terms from the *tProf* profiles in the following way: the *sProf* subjects are searched in the initiative associated *materias* tags and the *tProf* terms are searched in the *MaxUnit* content. Therefore, the underlying CAS query is as follows:

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

If we look at the performance of this new technique and all the others in Table 6.6, we can see that *CAS-mix* gets the best performance of the whole set of personalization techniques. Therefore, we can conclude that not very good results are obtained if we only use subjects as the profile information. However, we get a better performance if we use subjects together with terms, instead of only subjects.

Going further, if we compare each *sProf* Table 6.6 personalization technique results with their corresponding *tProf* Table 6.5 results, each *sProf* technique gets a lower performance, which again suggests it is better to use terms instead of subjects to build the user profiles. But, if we consider that the new developed *CAS-mix* personalization technique gets better performance than both *tProf* CAS approaches, not only for the highest user profile configuration parameters performance but also on average, even getting a lower standard deviation, it seems that the use of subjects

together with terms is a better approach that only use terms, as higher and more robust results are obtained.

6.3.3 Profiles based on subjects and terms

This user profile approach uses both subjects and terms. Concretely, and to remember, a given *stProf* user profile is composed by those subjects which best represent the user profile interests in a first level, having each of these subjects a set of its most representative terms in a second level. Therefore, this kind of user profile is a hybrid approach between the weighted concept and weighted keyword user profile representations, allowing to keep the concept abstraction but at the same time enriched by the terms fine-grained contribution.

As we have already seen in the previous section *CAS-mix* personalization technique results, the idea of mixing subjects and terms seems to be better than use any of them individually. In principle, as *stProf* profiles actually use terms but not subjects in the expansion process, they should at least partially solve the *sProf* profiles problems: on the one hand, the same amount of expansion terms will be used (this is only true in the **Fill** approaches). On the other hand, the expansion terms will match the document content, not only being semantically related to it as it happened when subjects are used. At the same time, as the expansion term weights were already multiplied by their corresponding subject weight, they already hold the subjects influence, which as we have already seen in the *CAS-mix* personalization technique contribute in some way. Let us see if that assumption is reinforced or not by the *stProf* user profiles.

Tables 6.7, 6.8, 6.9 and 6.10 are the obtained results for the Section 6.2.1 *stProf_add*, *stProf_max*, *stProf_addFill* and *stProf_maxFill* user profile approaches, respectively. Note that for Tables 6.9 (*stProf_addFill*) and 6.10 (*stProf_maxFill*), the *CAS-or* personalization technique with the user profile configuration parameters $k = 40$ and $l = 10$ has no available results, since its execution time is unacceptable even for experimental purposes.

Focusing on the results, we highlight the following conclusions:

- *stProf_max* and *stProf_maxFill* always obtain higher maximum and averaged performance results than *stProf_add* and *stProf_addFill*, respectively, for the

+m personalization techniques, while they obtain lower results under the *NQE, HRR, SRR* and *IRR* approaches (hereafter denoted as *base* personalization techniques). This fact shows how **max** approaches are more appropriated for personalization techniques which best avoid the *query-drift* problem, while **add** approaches are more appropriated for personalization techniques which partially avoid this problem.

- Both **Fill** approaches almost always get higher maximum results than *'noFill'* approaches, being always true if we compare *stProf_addFill-stProf_add* and *stProf_maxFill-stProf_max*, for *base* and *+m* personalization techniques. But when we focus on the average results, both previous **Fill** versus *'noFill'* profile comparisons get lower performance results considering the *base* personalization techniques, but higher values considering the *+m* techniques. This means that carefully selecting the user profile configuration parameters, **Fill** user profiles are always better, while if we are not sure about the suitability of these profile parameters, we still can trust on **Fill** user profiles for *+m* personalization techniques, but it could be better to use *'noFill'* user profiles for *base* personalization techniques.

- *CAS* approaches obtain very homogeneous results between the four different *stProf* user profiles, as it may be drawn from their low standard deviation values. Thus, there is no much difference between using whichever four user profile approaches.

- The absolute and averaged maximum performances are respectively obtained by *stProf_maxFill-HRR+m* and *stProf_addFill-CAS* user profile-personalization technique approaches, respectively, with $k = 40$, $l = 10$ and $p_0 = 0.99$ in the maximum value. The absolute maximum performance represents an improvement of 80.67% over the baseline.

6.3.4 All user profiles results and conclusions

In this section, we fuse and summarize all the above exposed results, for the six different proposed user profile approaches evaluated under the given evaluation framework. Tables 6.11 and 6.12 show these joined results.

Table 6.7: NDCG averaged values for the user profiles based on subjects and terms (*stProf_add*).

k	l	p_0	**NQE**	**HRR**	**SRR**	**IRR**	**NQE+m**	**HRR+m**	**SRR+m**	**IRR+m**	**CAS**	**CAS-or**
5	1	0.33	0.572	0.566	0.511	0.511	0.473	0.478	0.439	0.439	0.597	0.613
5	1	0.66	0.595	0.609	0.568	0.568	0.533	0.538	0.479	0.479	0.635	0.627
5	1	0.99	0.563	0.605	0.589	0.588	0.573	0.587	0.507	0.507	0.649	0.632
5	5	0.33	0.599	0.609	0.570	0.570	0.517	0.523	0.470	0.470	0.625	0.630
5	5	0.66	0.533	0.581	0.597	0.594	0.606	0.610	0.524	0.524	0.646	0.639
5	5	0.99	0.471	0.544	0.588	0.577	0.627	0.637	0.551	0.551	0.652	0.641
5	10	0.33	0.583	0.602	0.581	0.581	0.544	0.542	0.485	0.485	0.629	0.616
5	10	0.66	0.487	0.547	0.584	0.578	0.622	0.622	0.536	0.536	0.648	0.624
5	10	0.99	0.415	0.503	0.556	0.541	0.648	0.649	0.561	0.561	0.655	0.627
10	1	0.33	0.595	0.599	0.526	0.526	0.476	0.481	0.444	0.444	0.599	0.625
10	1	0.66	0.606	**0.626**	0.586	0.586	0.538	0.548	0.484	0.484	0.639	0.644
10	1	0.99	0.573	0.610	**0.605**	**0.603**	0.583	0.595	0.516	0.516	0.656	0.650
10	5	0.33	**0.610**	0.625	0.586	0.586	0.537	0.540	0.477	0.477	0.619	0.630
10	5	0.66	0.545	0.600	0.604	0.601	0.624	0.624	0.533	0.533	0.650	0.640
10	5	0.99	0.478	0.557	0.591	0.581	0.648	0.656	0.563	0.563	0.657	0.644
10	10	0.33	0.592	0.615	0.587	0.587	0.557	0.555	0.495	0.495	0.628	0.614
10	10	0.66	0.498	0.563	0.585	0.579	0.635	0.639	0.545	0.545	0.652	0.624
10	10	0.99	0.433	0.523	0.560	0.544	0.653	0.664	0.569	0.569	0.656	0.626
20	1	0.33	0.606	0.604	0.543	0.543	0.491	0.494	0.451	0.451	0.589	0.636
20	1	0.66	0.588	0.613	0.588	0.587	0.568	0.574	0.499	0.499	0.628	0.655
20	1	0.99	0.543	0.591	0.595	0.592	0.606	0.621	0.531	0.531	0.641	0.662
20	5	0.33	0.603	0.622	0.589	0.588	0.549	0.551	0.482	0.482	0.609	0.632
20	5	0.66	0.537	0.593	0.596	0.593	0.634	0.638	0.544	0.544	0.641	0.641
20	5	0.99	0.458	0.545	0.580	0.569	0.658	0.675	0.571	0.571	0.658	0.645
20	10	0.33	0.582	0.610	0.588	0.587	0.572	0.564	0.497	0.497	0.620	0.612
20	10	0.66	0.475	0.545	0.580	0.573	0.649	0.653	0.554	0.554	0.649	0.622
20	10	0.99	0.411	0.495	0.556	0.542	0.663	0.674	0.575	0.575	**0.659**	0.623
40	1	0.33	0.609	0.612	0.558	0.558	0.496	0.499	0.452	0.452	0.585	0.638
40	1	0.66	0.575	0.604	0.595	0.593	0.577	0.585	0.503	0.503	0.630	0.656
40	1	0.99	0.530	0.588	0.596	0.589	0.618	0.630	0.541	0.541	0.648	**0.662**
40	5	0.33	0.593	0.621	0.591	0.590	0.559	0.557	0.489	0.489	0.605	0.625
40	5	0.66	0.510	0.584	0.596	0.587	0.639	0.642	0.546	0.546	0.641	0.633
40	5	0.99	0.444	0.534	0.563	0.552	0.665	0.681	0.575	0.575	0.651	0.636
40	10	0.33	0.570	0.602	0.590	0.588	0.578	0.570	0.500	0.500	0.611	0.608
40	10	0.66	0.465	0.537	0.572	0.563	0.655	0.662	0.557	0.557	0.642	0.612
40	10	0.99	0.393	0.490	0.544	0.528	**0.673**	**0.685**	**0.580**	**0.580**	0.655	0.614
μ			0.534	0.580	0.578	0.573	0.590	0.596	0.517	0.517	0.635	0.632
σ			0.066	0.039	0.022	0.023	0.058	0.061	0.041	0.041	0.021	0.014
Baseline			0.388									

Table 6.8: NDCG averaged values for the user profiles based on subjects and terms (*stProf_max*).

k	l	p_0	**NQE**	**HRR**	**SRR**	**IRR**	**NQE+m**	**HRR+m**	**SRR+m**	**IRR+m**	**CAS**	**CAS-or**
5	1	0.33	0.575	0.576	0.519	0.519	0.479	0.484	0.444	0.444	0.602	0.622
5	1	0.66	0.595	0.615	0.573	0.573	0.543	0.551	0.485	0.485	0.641	0.639
5	1	0.99	0.556	0.605	0.591	0.590	0.584	0.598	0.515	0.515	0.656	0.644
5	5	0.33	0.594	0.615	0.587	0.587	0.545	0.551	0.486	0.486	0.635	0.636
5	5	0.66	0.513	0.572	0.601	0.594	0.622	0.629	0.543	0.543	0.653	0.646
5	5	0.99	0.439	0.530	0.586	0.572	0.649	0.658	0.569	0.569	0.660	0.648
5	10	0.33	0.565	0.597	0.586	0.585	0.562	0.561	0.494	0.494	0.632	0.615
5	10	0.66	0.472	0.544	0.584	0.573	0.641	0.643	0.549	0.549	0.651	0.627
5	10	0.99	0.403	0.499	0.556	0.536	0.656	0.664	0.574	0.574	**0.660**	0.629
10	1	0.33	0.599	0.602	0.541	0.541	0.487	0.492	0.448	0.448	0.604	0.629
10	1	0.66	0.592	0.621	0.590	0.589	0.553	0.566	0.494	0.494	0.642	0.645
10	1	0.99	0.553	0.603	**0.603**	**0.600**	0.596	0.610	0.525	0.525	0.652	0.653
10	5	0.33	0.592	**0.624**	0.588	0.588	0.558	0.561	0.489	0.489	0.621	0.632
10	5	0.66	0.506	0.570	0.600	0.593	0.635	0.641	0.546	0.546	0.647	0.638
10	5	0.99	0.433	0.522	0.579	0.563	0.656	0.667	0.571	0.571	0.652	0.641
10	10	0.33	0.561	0.604	0.587	0.585	0.575	0.573	0.500	0.500	0.624	0.607
10	10	0.66	0.455	0.527	0.575	0.564	0.646	0.653	0.552	0.552	0.650	0.618
10	10	0.99	0.389	0.481	0.549	0.527	0.664	0.680	0.580	0.580	0.656	0.619
20	1	0.33	**0.601**	0.605	0.549	0.549	0.504	0.511	0.456	0.456	0.600	0.642
20	1	0.66	0.561	0.598	0.582	0.580	0.573	0.584	0.508	0.508	0.639	0.655
20	1	0.99	0.516	0.575	0.593	0.587	0.616	0.634	0.540	0.540	0.651	**0.662**
20	5	0.33	0.565	0.601	0.580	0.579	0.567	0.569	0.496	0.496	0.617	0.620
20	5	0.66	0.474	0.541	0.576	0.567	0.648	0.655	0.554	0.554	0.648	0.629
20	5	0.99	0.402	0.496	0.553	0.536	0.669	0.680	0.577	0.577	0.655	0.632
20	10	0.33	0.532	0.576	0.573	0.570	0.587	0.583	0.506	0.506	0.618	0.603
20	10	0.66	0.418	0.495	0.546	0.532	0.661	0.665	0.558	0.558	0.645	0.608
20	10	0.99	0.356	0.440	0.521	0.498	0.672	0.688	0.583	0.583	0.656	0.610
40	1	0.33	0.589	0.596	0.549	0.549	0.515	0.518	0.461	0.461	0.589	0.642
40	1	0.66	0.536	0.586	0.577	0.574	0.587	0.595	0.515	0.515	0.629	0.655
40	1	0.99	0.478	0.544	0.576	0.568	0.624	0.640	0.547	0.547	0.647	0.657
40	5	0.33	0.548	0.583	0.570	0.568	0.573	0.576	0.500	0.500	0.603	0.609
40	5	0.66	0.439	0.512	0.559	0.550	0.651	0.658	0.556	0.556	0.636	0.619
40	5	0.99	0.373	0.469	0.540	0.522	0.675	0.688	0.581	0.581	0.647	0.620
40	10	0.33	0.507	0.549	0.567	0.563	0.592	0.588	0.511	0.511	0.604	0.594
40	10	0.66	0.400	0.476	0.538	0.523	0.663	0.670	0.561	0.561	0.638	0.598
40	10	0.99	0.341	0.430	0.505	0.483	**0.681**	**0.694**	**0.584**	**0.584**	0.645	0.600
μ			0.501	0.555	0.568	0.561	0.603	0.610	0.527	0.527	0.636	0.629
σ			0.080	0.054	0.024	0.029	0.057	0.059	0.041	0.041	0.020	0.018
Baseline			0.388									

Table 6.9: NDCG averaged values for the user profiles based on subjects and terms (*stProf_addFill*).

k	l	p_0	**NQE**	**HRR**	**SRR**	**IRR**	**NQE+m**	**HRR+m**	**SRR+m**	**IRR+m**	**CAS**	**CAS-or**
5	1	0.33	0.590	0.591	0.529	0.529	0.486	0.491	0.448	0.448	0.602	0.617
5	1	0.66	0.577	0.598	0.577	0.577	0.554	0.562	0.490	0.490	0.636	0.630
5	1	0.99	0.530	0.580	0.591	0.588	0.594	0.602	0.522	0.522	0.645	0.637
5	5	0.33	0.581	0.603	0.578	0.578	0.547	0.544	0.485	0.485	0.629	0.618
5	5	0.66	0.477	0.541	0.581	0.574	0.623	0.626	0.537	0.537	0.648	0.628
5	5	0.99	0.415	0.505	0.559	0.541	0.645	0.649	0.560	0.560	0.653	0.629
5	10	0.33	0.558	0.584	0.580	0.578	0.565	0.560	0.496	0.496	0.629	0.603
5	10	0.66	0.441	0.520	0.555	0.541	0.632	0.634	0.542	0.542	0.651	0.614
5	10	0.99	0.373	0.476	0.534	0.512	0.650	0.653	0.566	0.566	0.658	0.615
10	1	0.33	**0.626**	**0.634**	0.576	0.576	0.514	0.517	0.465	0.465	0.622	0.638
10	1	0.66	0.579	0.618	0.613	**0.611**	0.604	0.606	0.521	0.521	0.648	0.652
10	1	0.99	0.520	0.591	**0.615**	0.606	0.644	0.650	0.552	0.552	0.656	0.655
10	5	0.33	0.578	0.607	0.584	0.582	0.565	0.559	0.497	0.497	0.626	0.611
10	5	0.66	0.473	0.549	0.572	0.563	0.639	0.642	0.544	0.544	0.655	0.621
10	5	0.99	0.408	0.501	0.550	0.532	0.656	0.666	0.572	0.572	0.658	0.622
10	10	0.33	0.546	0.594	0.583	0.581	0.580	0.569	0.502	0.502	0.623	0.597
10	10	0.66	0.433	0.520	0.551	0.537	0.643	0.649	0.552	0.552	0.653	0.602
10	10	0.99	0.364	0.466	0.527	0.507	0.659	0.671	0.579	0.579	**0.660**	0.606
20	1	0.33	0.612	0.626	0.587	0.587	0.527	0.530	0.475	0.475	0.606	0.640
20	1	0.66	0.552	0.600	0.600	0.598	0.626	0.626	0.535	0.535	0.640	0.653
20	1	0.99	0.484	0.564	0.589	0.581	0.656	0.673	0.562	0.562	0.653	**0.659**
20	5	0.33	0.552	0.592	0.581	0.579	0.587	0.579	0.502	0.502	0.622	0.603
20	5	0.66	0.439	0.521	0.561	0.548	0.648	0.654	0.556	0.556	0.650	0.607
20	5	0.99	0.373	0.468	0.536	0.518	0.663	0.676	0.583	0.583	0.660	0.612
20	10	0.33	0.516	0.574	0.572	0.568	0.599	0.591	0.511	0.511	0.620	0.592
20	10	0.66	0.400	0.491	0.537	0.524	0.660	0.667	0.560	0.560	0.646	0.595
20	10	0.99	0.330	0.429	0.513	0.494	0.665	0.680	**0.585**	**0.585**	0.655	0.596
40	1	0.33	0.599	0.630	0.590	0.589	0.550	0.552	0.489	0.489	0.613	0.636
40	1	0.66	0.521	0.591	0.599	0.592	0.644	0.652	0.546	0.546	0.646	0.642
40	1	0.99	0.452	0.546	0.577	0.562	0.666	0.686	0.573	0.573	0.657	0.645
40	5	0.33	0.520	0.578	0.575	0.570	0.599	0.588	0.511	0.511	0.619	0.588
40	5	0.66	0.401	0.490	0.542	0.530	0.660	0.668	0.561	0.561	0.645	0.593
40	5	0.99	0.340	0.439	0.515	0.497	0.670	0.686	0.584	0.584	0.653	0.594
40	10	0.33	0.480	0.551	0.556	0.550	0.604	0.598	0.516	0.516	0.610	–
40	10	0.66	0.369	0.462	0.525	0.510	0.662	0.674	0.561	0.561	0.641	–
40	10	0.99	0.303	0.412	0.495	0.473	**0.674**	**0.693**	0.584	0.584	0.647	–
μ			0.481	0.546	0.564	0.555	0.616	0.620	0.534	0.534	0.640	0.620
σ			0.090	0.062	0.029	0.035	0.050	0.055	0.038	0.038	0.017	0.021
Baseline			0.388									

Table 6.10: NDCG averaged values for the user profiles based on subjects and terms (*stProf_maxFill*).

k	l	p_0	**NQE**	**HRR**	**SRR**	**IRR**	**NQE+m**	**HRR+m**	**SRR+m**	**IRR+m**	**CAS**	**CAS-or**
5	1	0.33	0.596	0.598	0.537	0.537	0.494	0.498	0.454	0.454	0.607	0.622
5	1	0.66	0.565	0.593	0.583	0.582	0.567	0.574	0.503	0.503	0.639	0.637
5	1	0.99	0.515	0.567	0.592	0.585	0.611	0.621	0.532	0.532	0.644	0.642
5	5	0.33	0.561	0.593	0.585	0.583	0.565	0.566	0.495	0.495	0.630	0.617
5	5	0.66	0.457	0.535	0.581	0.570	0.640	0.644	0.549	0.549	0.652	0.624
5	5	0.99	0.391	0.491	0.558	0.536	0.657	0.664	0.579	0.579	0.656	0.628
5	10	0.33	0.538	0.581	0.575	0.572	0.579	0.577	0.502	0.502	0.630	0.600
5	10	0.66	0.422	0.509	0.554	0.539	0.649	0.654	0.552	0.552	0.651	0.607
5	10	0.99	0.362	0.464	0.532	0.510	0.662	0.677	0.577	0.577	0.657	0.608
10	1	0.33	**0.612**	**0.633**	0.586	0.586	0.529	0.537	0.478	0.478	0.625	0.642
10	1	0.66	0.537	0.595	**0.606**	**0.602**	0.621	0.622	0.533	0.533	0.646	0.659
10	1	0.99	0.477	0.558	0.601	0.590	0.654	0.662	0.563	0.563	0.652	**0.660**
10	5	0.33	0.540	0.587	0.583	0.581	0.579	0.580	0.504	0.504	0.624	0.602
10	5	0.66	0.433	0.517	0.557	0.542	0.650	0.658	0.554	0.554	0.651	0.610
10	5	0.99	0.367	0.466	0.533	0.510	0.664	0.679	0.582	0.582	0.657	0.612
10	10	0.33	0.504	0.563	0.568	0.564	0.589	0.586	0.507	0.507	0.618	0.589
10	10	0.66	0.391	0.484	0.536	0.520	0.656	0.666	0.556	0.556	0.644	0.592
10	10	0.99	0.323	0.424	0.501	0.479	0.668	0.685	0.580	0.580	0.652	0.593
20	1	0.33	0.580	0.605	0.581	0.581	0.553	0.559	0.486	0.486	0.622	0.636
20	1	0.66	0.502	0.557	0.585	0.578	0.640	0.643	0.549	0.549	0.647	0.648
20	1	0.99	0.436	0.516	0.566	0.551	0.664	0.676	0.574	0.574	0.654	0.652
20	5	0.33	0.507	0.552	0.562	0.558	0.592	0.590	0.510	0.510	0.617	0.589
20	5	0.66	0.394	0.473	0.523	0.506	0.664	0.670	0.559	0.559	0.645	0.594
20	5	0.99	0.330	0.424	0.500	0.478	0.673	0.693	0.585	0.585	0.651	0.595
20	10	0.33	0.467	0.523	0.546	0.538	0.599	0.601	0.515	0.515	0.609	0.578
20	10	0.66	0.355	0.438	0.514	0.497	0.665	0.678	0.562	0.562	0.637	0.583
20	10	0.99	0.306	0.401	0.484	0.462	0.677	0.697	0.585	0.585	0.647	0.584
40	1	0.33	0.560	0.599	0.587	0.585	0.576	0.577	0.501	0.501	0.622	0.622
40	1	0.66	0.456	0.534	0.575	0.563	0.659	0.663	0.560	0.560	0.649	0.635
40	1	0.99	0.392	0.489	0.547	0.527	0.679	0.691	**0.587**	**0.587**	**0.658**	0.638
40	5	0.33	0.468	0.521	0.546	0.538	0.602	0.600	0.515	0.515	0.608	0.577
40	5	0.66	0.361	0.450	0.517	0.498	0.671	0.679	0.561	0.561	0.634	0.580
40	5	0.99	0.309	0.405	0.485	0.463	0.677	0.698	0.585	0.585	0.648	0.581
40	10	0.33	0.437	0.504	0.528	0.518	0.604	0.601	0.515	0.515	0.597	–
40	10	0.66	0.336	0.424	0.496	0.478	0.671	0.680	0.563	0.563	0.623	–
40	10	0.99	0.293	0.397	0.475	0.452	**0.683**	**0.701**	0.584	0.584	0.633	–
μ			0.447	0.516	0.550	0.538	0.627	0.635	0.542	0.542	0.637	0.613
σ			0.092	0.067	0.036	0.043	0.049	0.053	0.037	0.037	0.017	0.026
Baseline			0.388									

Table 6.11: NDCG maximum, average (μ) and std. (σ) performance values for the six developed user profile approaches under the evaluation framework. Original (non-personalized) NDCG value: 0.388. '*' character shows the best user profile approach for each personalization technique, and '$^+$' character shows the best personalization technique for a given user profile approach.

		NQE	**HRR**	**SRR**	**IRR**	**NQE+m**	**HRR+m**	**SRR+m**	**IRR+m**	**CAS**	**CAS-or**	**CAS-mix**
max	*tProf*	0.634*	0.652*	0.625*	0.620*	0.678	0.696$^+$	0.597*	0.597*	0.675*	0.668*	–
	sProf	0.588	0.603	0.577	0.572	0.632	0.645	0.552	0.552	0.552	0.578	0.679^{+*}
	stProf_add	0.610	0.626	0.605	0.603	0.673	0.685$^+$	0.580	0.580	0.659	0.662	–
	stProf_max	0.601	0.624	0.603	0.600	0.681	0.694$^+$	0.584	0.584	0.660	0.662	–
	stProf_addFill	0.626	0.634	0.615	0.611	0.674	0.693$^+$	0.585	0.585	0.660	0.659	–
	stProf_maxFill	0.612	0.633	0.606	0.602	0.683*	0.701^{+*}	0.587	0.587	0.658	0.660	–
μ	*tProf*	0.536*	0.593*	0.596*	0.589*	0.624	0.630	0.540	0.540	0.656^{+*}	0.645*	–
	sProf	0.513	0.565	0.560	0.553	0.566	0.572	0.501	0.501	0.532	0.565	0.668^{+*}
	stProf_add	0.534	0.580	0.578	0.573	0.590	0.596	0.517	0.517	0.635$^+$	0.632	–
	stProf_max	0.501	0.555	0.568	0.561	0.603	0.610	0.527	0.527	0.636$^+$	0.629	–
	stProf_addFill	0.481	0.546	0.564	0.555	0.616	0.620	0.534	0.534	0.640$^+$	0.620	–
	stProf_maxFill	0.447	0.516	0.550	0.538	0.627*	0.635*	0.542*	0.542*	0.637$^+$	0.613	–
σ	*tProf*	0.078	0.047	0.020	0.025	0.051	0.057	0.040	0.040	0.014^{+*}	0.014$^+$	–
	sProf	0.060*	0.035*	0.013*	0.014*	0.050	0.055	0.035*	0.035*	0.018	0.009*	0.007^{+*}
	stProf_add	0.066	0.039	0.022	0.023	0.058	0.061	0.041	0.041	0.021	0.014$^+$	–
	stProf_max	0.080	0.054	0.024	0.029	0.057	0.059	0.041	0.041	0.020	0.018$^+$	–
	stProf_addFill	0.090	0.062	0.029	0.035	0.050	0.055	0.038	0.038	0.017$^+$	0.021	–
	stProf_maxFill	0.092	0.067	0.036	0.043	0.049*	0.053*	0.037	0.037	0.017$^+$	0.026	–

Table 6.12: User profile parameters $k[-l]$–p_0 configuration for each maximum NDCG personalization technique-user profile performance, with '*' and '$^+$' characters meaning the same as in Table 6.11.

	NQE	**HRR**	**SRR**	**IRR**	**NQE+m**	**HRR+m**	**SRR+m**	**IRR+m**	**CAS**	**CAS-or**	**CAS-mix**
tProf	05-0.33*	10-0.33*	10-0.66*	10-0.66*	40-0.99	40-0.99$^+$	40-0.99*	40-0.99*	40-0.99*	05-0.99*	–
sProf	10-0.33	10-0.33	10-0.66	10-0.66	40-0.99	40-0.99	40-0.99	40-0.99	10-0.66	40-0.33	40-0.99^{+*}
stProf_add	10-05-0.33	10-01-0.66	10-01-0.99	10-01-0.99	40-10-0.99	40-10-0.99$^+$	40-10-0.99	40-10-0.99	20-10-0.99	40-01-0.99	–
stProf_max	20-01-0.33	10-05-0.33	10-01-0.99	10-01-0.99	40-10-0.99	40-10-0.99$^+$	40-10-0.99	40-10-0.99	05-05-0.99	20-01-0.99	–
stProf_addFill	10-01-0.33	10-01-0.33	10-01-0.99	10-01-0.66	40-10-0.99	40-10-0.99$^+$	20-10-0.99	20-10-0.99	10-10-0.99	20-01-0.99	–
stProf_maxFill	10-01-0.33	10-01-0.33	10-01-0.66	10-01-0.66	40-10-0.99*	40-10-0.99^{+*}	40-01-0.99	40-01-0.99	40-01-0.99	10-01-0.99	–

The first and main conclusion is the following: personalization always (except in very exceptional cases) helps the user to find relevant information faster and easier. Additionally, if we also carefully select the user profile parameters for any of the proposed user profiles, together with any personalization technique (note that there are good and not so good approaches for both), which is one of the goals of this chapter, we always get a quite good personalization improvement with respect to the non-personalized IRS performance (NDCG = 0.388) ranging from 42.27% to 80.67%.

The three main conclusions drawn from Table 6.11 are: 1) the best personalization technique in maximum and averaged NDCG values are clearly *HRR+m* and *CAS*, respectively; 2) the best user profile approach for maximum performance values is *tProf*, with the *NQE+m* and *HRR+m* exceptions, in which the *stProf_maxFill* profile is better. In the case of the averaged values, all the *+m* personalization techniques are the exceptions. And, 3) the new developed personalization technique *CAS-mix* is clearly the best approach considering *sProf* profiles. Besides, comparing it with all the other approaches, it achieves a considerable high maximum NDCG performance, after some of the *HRR+m* and *NQE+m* techniques configurations, but by far the highest averaged value and the lowest standard deviation of the whole comparison approaches.

Considering the previous conclusions, we may assume that most of the times the best user profile approach to use is the simpler *tProf*, instead of the bit more complicated *stProf_maxFill*, since the latter only achieves an improvement of 1.29% over the previous maximum performance obtained under the *tProf* user profile.

With respect to the used number of subjects or terms and the normalization value, i.e. $k[-l]-p_0$ parameters, Table 6.12 shows which user profile configuration maximizes performance. If we focus on the best personalization technique for all the user profile approaches ('+' character), the user profile configuration maximizing the performance is clearly 40-[10]-0.99, being *HRR+m* the personalization technique for all profiles except for the *sProf* profile, which is *CAS-mix*. Whereas, if we focus on the best user profile approach for all personalization techniques ('*' character), the user profile configuration maximizing the performance is composed by low values, such as $k = 5, 10$ or $p_0 = 0.33, 0.66$, for *NQE, HRR, SRR* and *IRR* techniques, and high values such as 40-[10]-0.99, for the rest of personalization techniques, except

Table 6.13: General NDCG maximum (max), average (μ) and deviation (σ) values for each of the six proposed user profile approaches.

	tProf	**sProf**	**stProf_add**	**stProf_max**	**stProf_addFill**	**stProf_maxFill**
max	0.644	0.594	0.628	0.629	0.634	0.633
μ	0.595	0.554	0.575	0.572	0.571	0.565
σ	0.039	0.030	0.039	0.042	0.043	0.045

for *CAS-or* which is 05-0.99. As we can see, these user profile configuration values basically depend on the given personalization technique, more than on the kind of user profile. Again, these conclusions verify that personalization techniques which solve the well-known *query-drift* problem get their maximum performance using the highest values of the user profile configuration parameters.

Last, Table 6.13 shows the general NDCG maximum (max), average (μ) and deviation (σ) values for each of the six proposed user profile approaches, i.e., this table shows the general averaged expected results for a given personalization technique for the six different user profile approaches. We may observe how the maximum and minimum *max* performances are achieved by the *tProf* and *sProf* user profiles, respectively, while the *stProf* profiles obtain relatively good values generally increasing while we move to the right in the table. We also observe how the highest average (μ) value is achieved by the *tProf* approach, with a low deviation (σ) value. Meanwhile, the lowest deviation value is achieved by the *sProf* approach, but with a much lower average value than *tProf*. Considering the four *stProf* approaches, we may observe a gradual decrease and increase in the average and deviation values, respectively, following the order of these profiles in the table. This situation indicates that within these user profiles, the further to the right in the table, they achieve more disparate personalization results (higher and lower), so more attention need to be paid to the selection of the right user profile configuration. The fact of having the maximum experimental evaluation performance with *stProf_maxFill* approach confirms this last conclusion.

Considering all the results, could it be concluded that we stand up for the *tProf* profile? Not necessarily. From a user perspective and considering not very small profiles, a *stProf* profile is much easier to understand than a *tProf* profile, since abstract concepts contain more semantics than isolated terms. It is also true that the *stProf* profile with two levels (concepts and terms) could be exploited by a given

personalization technique to improve its performance, e.g., easily selecting parts of the user profile which suit more to the query (particularly helpful for heterogeneous profiles). Thus, depending on the application and the used personalization technique, a trade-off decision between pure performance or more expressiveness of the user profile must be taken.

6.4 Conclusions and future work

Since user profiles are very important for personalization, in this chapter we have presented six different user profile representation approaches based on content. Although these content-based user profiles do not represent real users, which is their main disadvantage, they are a suitable approach for representing user interests while having many other advantages, such as: they are perfect to allow personalization in privacy-constrained environments, since they do not collect any personal information. Derived from this last fact, they do not place any burden on the user at all, also not needing any complex user gathering information process or tools to be installed on the client side. They are also more easy and less expensive to be maintained, since a low number of them are going to exist (as many as collection categories, which are much less than possible IRS users), they can be updated very easily being only necessary with any addition or modification of the content (e.g. with the possibility to be updated only at wee hours). They can be stored on the server, avoiding some network traffic and, more important, to send personal information with the involved risks. Additionally, they are perfect to be used in evaluation frameworks based on contextual simulations. They even could be used as a real user first version profile ('cold-start'), etc.

We have developed a new way to build the user profiles based on terms, which improves our previous way to build them, being at the same time, also compatible with other sources to build the user profiles. Next, we have presented the user profiles based on subjects, which are manually assigned from a thesaurus to the documents initiatives by documentalists, being these subjects considered as concepts. These last user profiles showed a relatively poor performance in comparison with the user profiles based on terms. Therefore, we tried to improve their results with different approximations, until we ended up developing a new personalization

technique which uses both subjects and terms, and gets quite good performance results. Finally, inspired by the previous personalization technique, we have proposed a hybrid approach between the two previous user profile approaches (including four variations), thus having a two level user profile representation, where the first level is represented by subjects and the second level by the terms representing these subjects.

We have performed evaluation experiments including ten different personalization techniques (eleven in the case of user profiles based on subjects) and a wide range of user profile configurations, for all the proposed user profile approaches. We have obtained very good results, which in the best case reach up to *80.67%* of improvement, with respect to the original non-personalized IRS. Additionally, we have demonstrated that most of the times the use of a simple user profile based on terms is enough to get good personalized results. Anyway, having a user profile with some structure and abstract concepts may help both, users to better understand their own profiles, and also some personalization techniques which may exploit this richer representation. In fact, these user profiles with concepts are specially suitable for some IR subfields as multilingual IR, as for example [50] authors reveal as a possible future work for this very recent article.

As this chapter research future work, we would like to incorporate some extra information to the user profiles, such as localization or temporal information. We also would like to develop some personalization techniques in order to better exploit the hierarchy of the proposed user profiles based on subjects and terms. A definitive future work we would like to carry out is to use these proposed content-based user profiles to include personalization in privacy-constrained environments, such as in the Andalusian Parliament.

Part IV

Conclusions

Chapter 7

Conclusions and Future work

7.1 Conclusions

This section presents this thesis general conclusions. It should be noted that more specific and extended conclusions were previously given, at the end of each corresponding research contributions chapter.

Chapters 2 and 3 present a general overview and the state-of-the-art of personalized and structured IR, respectively. These chapters are helpful to describe the knowledge field to which this thesis belongs to. The different areas comprising this knowledge field along with the corresponding main bibliography references are exposed, with the aim of understanding and having a general vision of this knowledge field, to know where our research contributions are framed and the problems we try to solve or to improve.

Chapter 4 shows how a good design of the personalization techniques, which make use of the user profile information to retrieve results closer to the user, may considerably impact on the IRS output performance in order to best satisfy the user information needs. We have concretely faced the development of personalization techniques for the retrieval of structured (XML) documents, which is a research area still relatively unexplored. Although these techniques have been developed for XML IR, implying some extra challenges compared to the development of techniques for non-structured IR, they can be easily adapted to work with plain documents. In this chapter, we also present the common experimental components used in most of

the evaluations of this thesis, such as the XML document collection and the carried out user study.

We have developed a wide set of 13 different personalization techniques, which use the user profile information in the three common scenarios (or combinations of them) where personalization can be implemented: before, within and after the search is performed. Some design aspects of these personalization techniques that stand out are the use of two different lists of results in the reranking strategies, the modification of the search engine retrieval model (not very frequent) or the use of CAS queries for personalization, which as far as we know, nobody else has used for personalization purposes. Most of these techniques, in various ways and at different levels, solve the well-known *query-drift* problem.

We have used the NDCG and RI evaluation metrics within the experimental evaluations. It has been necessary to adapt NDCG to properly work with structured documents, by the integration of an overlap degree and a structural normalization between the structural units of the retrieved results and the user study relevance assessments. As the evaluation results show all the proposed strategies significantly improve the baseline (not personalized) results, reaching up to 84.5% and 71.25% maximum and averaged NDCG improvements, by the corresponding personalization techniques respectively.

Chapter 5 proposes an Automatic Strategy for Personalized Information Retrieval systems Evaluation denominated ASPIRE, which aims to join the advantages of the system-centred and user-centred evaluation approaches, i.e., to produce repeatable, comparable and generalizable results while including the user context within the retrieval process. ASPIRE is framed under contextual simulations, but it has several advantages over them, e.g., its key ability to generate automatic relevance assessments.

ASPIRE pretends to be an alternative but not a replacement of the costly user studies, turning the difficult but at the same time crucial personalization evaluation process into an easy, fast and low effort and cost process, with the only requisite of having at least a partially classifiable document collection.

The reliability and robustness of ASPIRE have been thoroughly tested by the comprehensive comparison (three retrieval models including structured and non-structured ones) between its results and those obtained from the carried out user

study. Figure 5.4 is very clarifying, showing Pearson correlation values higher than 0.914 between the ASPIRE and the user study sets of results. ASPIRE is specially useful to select the best personalization technique from a set of them, or the best user profile configuration for a given personalization technique. This last feature has also been extensively tested, showing high Kendall τ correlation values between the rankings obtained by ASPIRE and the user study.

Chapter 6 shows our six developed different approaches to represent content-based user profiles information. These profiles are based on the document collection terms and subjects. Although content-based user profiles do not represent real users, which is their main disadvantage, they are a suitable approach for representing user interests. They also have several advantages, such as, they allow personalization in privacy-constrained environments, they do not collect personal information which is a problem most of the times, or they are much easier to build and maintain, among many others.

We first present a user profile approach based on terms, which obtains quite good results. Then, we present a second approach based on subjects, which obtains lower performance than the previous one based on terms. Trying to improve its results we even ended up developing a new personalization technique, which mixes subjects and terms obtaining good results. Finally, we present four user profiles based on subjects and terms approaches, one of which gets the highest performance of the six proposed approaches.

The evaluation included ten different personalization techniques (eleven for the user profiles based on subjects) and a broad spectrum of user profile configurations. We show how with the use of a simple user profile based on terms, most of the times quite good performance results are obtained. But, the use of more sophisticated user profiles having some internal structure and abstract concepts are sometimes preferred, which also obtain very good results. As the final conclusion, we think a trade-off decision between simplicity or more expressiveness of the user profile must be taken, since according to our evaluation results the performance is quite similar.

7.2 Future work

This section presents this thesis possible future work directions. It should be noted that future works were also previously proposed, at the end of each corresponding research contributions chapter.

Regarding to personalization techniques, we would like to be able to select the best technique and/or tune their parameters and those of the user profiles based on some retrieval factors, such as the query characteristics or the user full context. Another possibility would be to incorporate some novelty or diversity factors, in order to include hot recent results or to discover new information that would not be discovered with a user profile too specialized. New personalization techniques may be developed in order to better exploit the hierarchy of the user profiles based on subjects and terms. Going further, another possible future work, more likely within very specialized environments, would be to also exploit the user structural preferences in the retrieval process.

In the case of the personalization evaluation step, we would like to explore some other criteria for the automatic relevance assessments generation. We also would like to extend ASPIRE in order to consider the user-IRS interactions within the automatic evaluation process.

With respect to user profiles, we would like to include and exploit more user context information within the profile, such as maybe personal data, device, localization, etc. We also would like to include a temporal component into the user profile, in order to discern the short-term and long-term interests and preferences, then being able to provide more accurate results.

In general, it would be also interesting to make the IRSs interfaces dynamically configurable, which could be another feature to store in the user profile, and to study how much this feature would increase the user final satisfaction with the IRS. Of course, one of the main and most desirable future works would be to introduce personalization in the Andalusian Parliament or any other place (wherever it is possible). The implementation in a real scenario would give us the opportunity to make available to real people the most part of this thesis contributions, and even to analyse all users provided data to keep improving our research results.

7.3 List of publications

The research contributions and results presented in this thesis have also been published in the following list of publications:

1. L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, and E. Vicente-López. *Personalización y Evaluación XML mediante la Simulación de Perfiles de Usuario y Juicios de Relevancia*. Proceedings of the 2nd Spanish Conference on Information Retrieval (CERI'12), pages 211–222, 2012.

2. 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 Annual ACM Symposium on Applied Computing (SAC'13), pages 872–877, 2013.

3. L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, and E. Vicente-López. *Using Personalization to Improve XML Retrieval*. IEEE Transactions on Knowledge and Data Engineering (TKDE), 26(5):1280–1292, 2014.

4. E. Vicente-López, L.M. de Campos, J.M. Fernández-Luna, J.F. Huete, A. Tagua-Jiménez, and C. Tur-Vigil. *An Automatic Methodology to Evaluate Personalized Information Retrieval Systems*. User Modeling and User-Adapted Interaction (UMUAI), to appear, 2015.

5. E. Vicente-López, L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. *Perfiles de Usuario Simulados basados en Materias y Términos para la Personalización*. Proceedings of the 3rd Spanish Conference on Information Retrieval (CERI'14), pages 1–12, 2014.

6. E. Vicente-López, L.M. de Campos, J.M. Fernández-Luna, and J.F. Huete. *Personalization of Parliamentary Document Retrieval using different User Profiles*. Proceedings of the 2nd International Workshop on Personalization in eGovernment Services and Applications (PeGOV'14), in conjunction with the 22nd Conference on User Modeling, Adaptation and Personalization (UMAP'14), pages 28–37, 2014.

The following reference is a publication partially related with this thesis content, concretely a web interface for the Garnata IRS applied to the Andalusian Parliament:

1. L.M. de Campos, A. Ching, J.M. Fernández-Luna, J.F. Huete, A. Tagua-Jiménez, C. Tur-Vigil, and E. Vicente-López. *Seda: un motor de búsqueda para las colecciones (XML) del Parlamento de Andalucía*. Proceedings of the 3rd Spanish Conference on Information Retrieval (CERI'14), pages 97–108, 2014.

References

[1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *Handheld and ubiquitous computing*, pages 304–307. Springer, 1999. → *Cited on page 98.*

[2] J.-W. Ahn, P. Brusilovsky, D. He, J. Grady, and Q. Li. Personalized web exploration with task models. In *Proceedings of the 17th international conference on World Wide Web*, pages 1–10. ACM, 2008. → *Cited on page 14.*

[3] J. Allan. Hard track overview in trec 2003 high accuracy retrieval from documents. Technical report, DTIC Document, 2005. → *Cited on page 30.*

[4] S. Amer-Yahia, I. Fundulaki, P. Jain, and L. Lakshmanan. Personalizing xml text search in piment. In *Proceedings of the 31st international conference on Very large data bases*, pages 1310–1313. VLDB Endowment, 2005. → *Cited on page 63.*

[5] S. Amer-Yahia, I. Fundulaki, and L. V. Lakshmanan. Personalizing xml search in pimento. In *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*, pages 906–915. IEEE, 2007. → *Cited on page 64.*

[6] L. Azzopardi, M. De Rijke, and K. Balog. Building simulated queries for known-item topics: an analysis using six european languages. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 455–462. ACM, 2007. → *Cited on page 107.*

[7] R. Baeza-Yates and B. Ribeiro-Neto. *Modern information retrieval*, volume 463. ACM press New York, 1999. → *Cited on page 38.*

[8] B. Bahmani, K. Chakrabarti, and D. Xin. Fast personalized pagerank on mapreduce. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 973–984. ACM, 2011. → *Cited on page 24.*

[9] P. Bailey, N. Craswell, I. Soboroff, P. Thomas, A. P. de Vries, and E. Yilmaz. Relevance assessment: are judges exchangeable and does it matter. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 667–674. ACM, 2008. → *Cited on page 122.*

[10] R. Barrett, P. P. Maglio, and D. C. Kellem. How to personalize the web. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pages 75–82. ACM, 1997. → *Cited on page 18.*

[11] N. J. Belkin. Some (what) grand challenges for information retrieval. *ACM SIGIR Forum*, 42(1):47–54, 2008. → *Cited on pages 4 and 13.*

[12] P. Borlund. The iir evaluation model: a framework for evaluation of interactive information retrieval systems. *Information research*, 8(3), 2003. → *Cited on pages 29, 33, 74, 98, and 107.*

[13] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 25–32. ACM, 2004. → *Cited on page 106.*

[14] K. Byström and K. Järvelin. Task complexity affects information seeking and use. *Information processing & management*, 31(2):191–213, 1995. → *Cited on page 32.*

[15] S. Calegari and G. Pasi. Ontology-based information behaviour to improve web search. *Future Internet*, 2(4):533–558, 2010. → *Cited on page 20.*

[16] C. Carpineto and G. Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1–50, 2012. → *Cited on pages 23 and 83.*

[17] B. Carterette, V. Pavlu, E. Kanoulas, J. A. Aslam, and J. Allan. Evaluation over thousands of queries. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 651–658. ACM, 2008. → *Cited on pages 106 and 107.*

[18] B. Carterette and I. Soboroff. The effect of assessor error on ir system evaluation. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 539–546. ACM, 2010. → *Cited on page 122.*

[19] H. Chang, D. Cohn, and A. K. McCallum. Learning to create customized authority lists. In *Proceedings of the 2000 International Conference on Machine Learning (ICML'00)*, pages 127–134, 2000. → *Cited on page 24.*

[20] L. Chen and K. Sycara. Webmate: a personal agent for browsing and searching. In *Proceedings of the second international conference on Autonomous agents*, pages 132–139. ACM, 1998. → *Cited on pages 13 and 19.*

[21] G. Chernishev. Personalization of xml text search via search histories. In *SYRCoDIS*, 2008. → *Cited on page 63.*

[22] P. R. Chesnais, M. J. Mucklo, and J. A. Sheena. The fishwrap personalized news system. In *Community Networking, 1995. Integrated Multimedia Services to the Home., Proceedings of the Second International Workshop on*, pages 275–282. IEEE, 1995. → *Cited on page 13.*

[23] Y. Chiaramella. Information retrieval and structured documents. In *Lectures on information retrieval*, pages 286–309. Springer, 2001. → *Cited on page 4.*

[24] D. N. Chin. Empirical evaluation of user models and user-adapted systems. *User modeling and user-adapted interaction*, 11(1-2):181–194, 2001. → *Cited on pages 33, 74, and 107.*

[25] P.-A. Chirita, C. S. Firan, and W. Nejdl. Summarizing local context to personalize global web search. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 287–296. ACM, 2006. → *Cited on page 26.*

[26] P.-A. Chirita, C. S. Firan, and W. Nejdl. Personalized query expansion for the web. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 7–14. ACM, 2007. → *Cited on page 23.*

[27] C. W. Cleverdon. The significance of the cranfield tests on index languages. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 3–12. ACM, 1991. → *Cited on page 3.*

[28] C. W. Cleverdon, J. Mills, and E. Keen. Factors determining the performance of indexing systems,(volume 1: Design). *Cranfield: College of Aeronautics*, 1966. → *Cited on pages 28, 52, and 98.*

[29] O. Craveiro, J. Macedo, and H. Madeira. Query expansion with temporal segmented texts. In *Advances in Information Retrieval*, pages 612–617. Springer, 2014. → *Cited on page 23.*

[30] W. B. Croft, S. Cronen-Townsend, and V. Lavrenko. Relevance feedback and personalization: A language modeling perspective. In *DELOS Workshop: Personalisation and Recommender Systems in Digital Libraries*, volume 3, 2001. → *Cited on page 13.*

[31] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306. ACM, 2002. → *Cited on page 23.*

[32] M. Daoud, L. Tamine, and M. Boughanem. A contextual evaluation protocol for a session-based personalized search. In *Workshop on Contextual Information Access, Seeking and Retrieval Evaluation (In conjunction with European Conference on Information retrieval-ECIR), Toulouse, France, Springer*, 2009. → *Cited on page 105.*

[33] M. Daoud, L. Tamine-Lechani, and M. Boughanem. Learning user interests for a session-based personalized search. In *Proceedings of the second international*

symposium on Information interaction in context, pages 57–64. ACM, 2008. → *Cited on pages 21 and 135.*

[34] L. M. de Campos, J. M. Fernández-Luna, and J. F. Huete. Using context information in structured document retrieval: an approach based on influence diagrams. *Information processing & management*, 40(5):829–847, 2004. → *Cited on page 56.*

[35] 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. In *Advances in Information Retrieval*, pages 215–229. Springer, 2005. → *Cited on page 56.*

[36] 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. In *Flexible Query Answering Systems*, pages 617–628. Springer, 2009. → *Cited on page 64.*

[37] 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 & management*, 46(5):514–532, 2010. → *Cited on page 44.*

[38] 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. In *Advances in Focused Retrieval*, pages 39–45. Springer, 2009. → *Cited on page 56.*

[39] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and C. J. Martín-Dancausa. A content-based approach to relevance feedback in xml-ir for content and structure queries. In *KDIR*, pages 418–427, 2010. → *Cited on page 64.*

[40] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, C. J. Martin-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. → *Cited on page 56.*

[41] L. M. De Campos, J. M. Fernández-Luna, J. F. Huete, and A. E. Romero. 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)*, pages 1024–1031, 2006. → *Cited on page 37.*

[42] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, and E. Vicente-López. Using personalization to improve xml retrieval. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 26(5):1280–1292, 2014. → *Cited on page 6.*

[43] A. Díaz, A. García, and P. Gervás. User-centred versus system-centred evaluation of a personalization system. *Information Processing & Management*, 44(3):1293–1307, 2008. → *Cited on pages 34 and 98.*

[44] C. Ding and J. C. Patra. User modeling for personalized web search with self-organizing map. *Journal of the American Society for information Science and Technology*, 58(4):494–507, 2007. → *Cited on page 29.*

[45] Z. Dou, R. Song, and J.-R. Wen. A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on World Wide Web*, pages 581–590. ACM, 2007. → *Cited on page 13.*

[46] D. Elsweiler, D. E. Losada, J. C. Toucedo, and R. T. Fernández. Seeding simulated queries with user-study data forpersonal search evaluation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 25–34. ACM, 2011. → *Cited on page 107.*

[47] S. French. *Decision theory: an introduction to the mathematics of rationality.* Halsted Press, 1986. → *Cited on page 55.*

[48] S. Gauch, M. Speretta, A. Chandramouli, and A. Micarelli. User profiles for personalized information access. In *The adaptive web*, pages 54–89. Springer, 2007. → *Cited on pages 15 and 19.*

[49] G. Gentili, A. Micarelli, and F. Sciarrone. Infoweb: An adaptive information filtering system for the cultural heritage domain. *Applied Artificial Intelligence*, 17(8-9):715–744, 2003. → *Cited on page 19.*

[50] M. R. Ghorab, S. Lawless, A. O'Connor, and V. Wade. Does personalization benefit everyone in the same way? multilingual search personalization for english vs. non-english users. In *Proceedings of the Joint Workshop on Personalized Information Access (PIA 2014), in conjunction with the 22nd conference on User Modeling, Adaptation and Personalization (UMAP 2014)*, pages 40–47. Springer, 2014. → *Cited on page 157.*

[51] M. R. Ghorab, D. Zhou, A. O'Connor, and V. Wade. Personalised information retrieval: survey and classification. *User Modeling and User-Adapted Interaction*, 23(4):381–443, 2013. → *Cited on page 13.*

[52] G. J. Hahm, M. Y. Yi, J. H. Lee, and H. W. Suh. A personalized query expansion approach for engineering document retrieval. *Advanced Engineering Informatics*, 2014. → *Cited on page 23.*

[53] D. Harman. Overview of the fourth text retrieval conference (trec-4). In *The Fourth Text REtrieval Conference (TREC-4)*, pages 1–24, 1995. → *Cited on page 30.*

[54] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 15(4):784–796, 2003. → *Cited on page 24.*

[55] E. Herrera-Viedma, G. Pasi, and F. Crestani. *Soft Computing in Web Information Retrieval*, volume 197. Springer Berlin Heidelberg, 2006. → *Cited on page 14.*

[56] 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):1–24, 2010. → *Cited on page 65.*

[57] W. Hsu, M. L. Lee, and X. Wu. Path-augmented keyword search for xml documents. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 526–530. IEEE, 2004. → *Cited on page 65.*

[58] iGoogle. http://www.google.es/ig. (last time available on November 2013). → *Cited on page 17.*

[59] P. Ingwersen. Cognitive perspectives of information retrieval interaction: elements of a cognitive ir theory. *Journal of documentation*, 52(1):3–50, 1996. → *Cited on page 28.*

[60] IOS Press - STM Publishing house. http://iospress.metapress.com/home/main.mpx. (last access on December 2014). → *Cited on page 14.*

[61] B. J. Jansen, A. Spink, and T. Saracevic. Real life, real users, and real needs: a study and analysis of user queries on the web. *Information processing & management*, 36(2):207–227, 2000. → *Cited on page 76.*

[62] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002. → *Cited on pages 33 and 54.*

[63] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, pages 271–279. ACM, 2003. → *Cited on page 24.*

[64] J. Kamps, J. Pehcevski, G. Kazai, M. Lalmas, and S. Robertson. Inex 2007 evaluation measures. In *Focused access to XML documents*, pages 24–33. Springer, 2008. → *Cited on pages 55 and 65.*

[65] G. Kazai and M. Lalmas. *INEX 2005 evaluation measures*. Springer, 2006. → *Cited on page 36.*

[66] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *ACM SIGIR Forum*, 37(2):18–28, 2003. → *Cited on page 13.*

[67] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999. → *Cited on page 24.*

[68] A. Kobsa. Privacy-enhanced web personalization. In *The adaptive web*, pages 628–670. Springer, 2007. → *Cited on page 14.*

[69] M. Lalmas. Xml retrieval. *Synthesis Lectures on Information Concepts, Retrieval and Services*, 1(1):1–111, 2009. → *Cited on pages 36, 42, and 49.*

[70] E. L.-C. Law, T. Klobučar, and M. Pipan. *User effect in evaluating personalized information retrieval systems*. Springer, 2006. → *Cited on pages 15 and 133.*

[71] H. Lieberman. Letizia: An agent that assists web browsing. *IJCAI*, 1:924–929, 1995. → *Cited on pages 13 and 18.*

[72] H. Lieberman, N. Van Dyke, and A. Vivacqua. Let's browse: a collaborative browsing agent. *Knowledge-Based Systems*, 12(8):427–431, 1999. → *Cited on page 18.*

[73] F. Liu, C. Yu, and W. Meng. Personalized web search for improving retrieval effectiveness. *Knowledge and Data Engineering, IEEE transactions on*, 16(1):28–40, 2004. → *Cited on pages 14 and 29.*

[74] C. Macdonald and I. Ounis. Expertise drift and query expansion in expert search. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 341–350. ACM, 2007. → *Cited on page 23.*

[75] T. W. Malone, K. R. Grant, F. A. Turbak, S. A. Brobst, and M. D. Cohen. Intelligent information-sharing systems. *Communications of the ACM*, 30(5):390–402, 1987. → *Cited on page 13.*

[76] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008. → *Cited on pages 11, 13, 36, 46, and 48.*

[77] Y. Mass and M. Mandelbrod. Relevance feedback for xml retrieval. In *Advances in XML Information Retrieval*, pages 303–310. Springer, 2005. → *Cited on page 64.*

[78] N. Matthijs and F. Radlinski. Personalizing web search using long term browsing history. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 25–34. ACM, 2011. → *Cited on page 26.*

[79] 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. → *Cited on pages 26, 67, and 68.*

[80] L. Meister, O. Kurland, and I. G. Kalmanovich. Re-ranking search results using an additional retrieved list. *Information retrieval*, 14(4):413–437, 2011. → *Cited on page 26.*

[81] A. Micarelli, F. Gasparetti, F. Sciarrone, and S. Gauch. Personalized search on the world wide web. In *The adaptive web*, pages 195–230. Springer, 2007. → *Cited on page 14.*

[82] A. Micarelli and F. Sciarrone. Anatomy and empirical evaluation of an adaptive web-based information filtering system. *User Modeling and User-Adapted Interaction*, 14(2-3):159–200, 2004. → *Cited on page 19.*

[83] C. E. Mooers. Coding, information retrieval, and the rapid selector. *American Documentation*, 1(4):225–229, 1950. → *Cited on page 3.*

[84] J. Mostafa, S. Mukhopadhyay, and M. Palakal. Simulation studies of different dimensions of users' interests and their impact on user modeling and information filtering. *Information Retrieval*, 6(2):199–223, 2003. → *Cited on page 104.*

[85] A. Moukas. Amalthaea information discovery and filtering using a multiagent evolving ecosystem. *Applied Artificial Intelligence*, 11(5):437–457, 1997. → *Cited on page 19.*

[86] MyYahoo! https://my.yahoo.com/. (last access on December 2014). → *Cited on page 17.*

[87] L. Nie, B. D. Davison, and X. Qi. Topical link analysis for web search. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 91–98. ACM, 2006. → *Cited on page 24.*

[88] T. D. Nielsen and F. V. Jensen. *Bayesian networks and decision graphs*. Springer, 2009. → *Cited on page 55.*

[89] R. Nuray and F. Can. Automatic ranking of retrieval systems in imperfect environments. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 379–380. ACM, 2003. → *Cited on page 106.*

[90] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999. → *Cited on page 24.*

[91] S. Pal, M. Mitra, and J. Kamps. Evaluation effort, reliability and reusability in xml retrieval. *Journal of the American Society for Information Science and Technology*, 62(2):375–394, 2011. → *Cited on pages 78 and 106.*

[92] G. Paltoglou and M. Thelwall. A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1386–1395. Association for Computational Linguistics, 2010. → *Cited on page 46.*

[93] H. Pan. Relevance feedback in xml retrieval. In *Current Trends in Database Technology-EDBT 2004 Workshops*, pages 187–196. Springer, 2005. → *Cited on page 64.*

[94] J. Parapar, M. A. Presedo-Quindimil, and A. Barreiro. Score distributions for pseudo relevance feedback. *Information Sciences*, 273(0):171–181, 2014. → *Cited on page 23.*

[95] G. Pasi. Issues in personalizing information retrieval. *IEEE Intelligent Informatics Bulletin*, 11(1):3–7, 2010. → *Cited on pages 15, 17, 21, and 133.*

[96] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988. → *Cited on page 55.*

[97] M. Perkowitz and O. Etzioni. Towards adaptive web sites: Conceptual framework and case study. *Artificial intelligence*, 118(1):245–275, 2000. → *Cited on page 13.*

[98] D. Petrelli. On the role of user-centred evaluation in the advancement of interactive information retrieval. *Information processing & management*, 44(1):22–38, 2008. → *Cited on pages 31 and 33.*

[99] M. F. Porter. An algorithm for suffix stripping. *Program: electronic library and information systems*, 14(3):130–137, 1980. → *Cited on page 39.*

[100] V. Ramesh, R. L. Glass, and I. Vessey. Research in computer science: an empirical study. *Journal of systems and software*, 70(1):165–176, 2004. → *Cited on page 28.*

[101] S. E. Robertson and K. S. Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146, 1976. → *Cited on page 48.*

[102] S. E. Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3. *NIST SPECIAL PUBLICATION SP*, pages 109–109, 1995. → *Cited on pages 49 and 108.*

[103] T. Sakai and N. Kando. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Information Retrieval*, 11(5):447–470, 2008. → *Cited on page 112.*

[104] T. Sakai, T. Manabe, and M. Koyama. Flexible pseudo-relevance feedback via selective sampling. *ACM Transactions on Asian Language Information Processing (TALIP)*, 4(2):111–135, 2005. → *Cited on page 83.*

[105] G. Salton, A. Wong, and C.-S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975. → *Cited on pages 48 and 108.*

[106] M. Sanderson and I. Soboroff. Problems with kendall's tau. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 839–840. ACM, 2007. → *Cited on page 106.*

[107] E. Santos Jr, Q. Zhao, H. Nguyen, and H. Wang. Impacts of user modeling on personalization of information retrieval: an evaluation with human intelligence analysts. In *Proceedings of the fourth workshop on the evaluation of adaptive systems (held in conjunction with the 10th International Conference on User Modeling (UM-05)), Edinburgh, UK*, pages 27–36. Citeseer, 2005. → *Cited on page 32.*

[108] T. Saracevic. Relevance: A review of and a framework for the thinking on the notion in information science. *Journal of the American Society for Information Science*, 26(6):321–343, 1975. → *Cited on page 122.*

[109] R. Schenkel and M. Theobald. Feedback-driven structural query expansion for ranked retrieval of xml data. In *Advances in Database Technology-EDBT 2006*, pages 331–348. Springer, 2006. → *Cited on page 65.*

[110] R. Schenkel and M. Theobald. Structural feedback for keyword-based xml retrieval. In *Advances in Information Retrieval*, pages 326–337. Springer, 2006. → *Cited on page 65.*

[111] R. D. Shachter. Evaluating influence diagrams. *Operations research*, 34(6):871–882, 1986. → *Cited on page 56.*

[112] J. Shavlik, S. Calcari, T. Eliassi-Rad, and J. Solock. An instructable, adaptive interface for discovering and monitoring information on the world-wide web. In *Proceedings of the 4th international conference on Intelligent user interfaces*, pages 157–160. ACM, 1998. → *Cited on page 17.*

[113] X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 824–831. ACM, 2005. → *Cited on pages 18, 23, and 26.*

[114] L. Shou, H. Bai, K. Chen, and G. Chen. Supporting privacy protection in personalized web search. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 26(2):453–467, 2012. → *Cited on page 134.*

[115] A. Sieg, B. Mobasher, and R. Burke. Web search personalization with ontological user profiles. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 525–534. ACM, 2007. → *Cited on pages 20, 31, 105, 114, 125, 131, 134, and 135.*

[116] I. Soboroff, C. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 66–73. ACM, 2001. → *Cited on pages 106 and 107.*

[117] Y. Song, H. Wang, and X. He. Adapting deep ranknet for personalized search. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 83–92. ACM, 2014. → *Cited on pages 24 and 25.*

[118] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments: Part 1. *Information Processing & Management*, 36(6):779–808, 2000. → *Cited on page 48.*

[119] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information Processing & Management*, 36(6):809–840, 2000. → *Cited on page 48.*

[120] A. Spink, S. Ozmutlu, H. C. Ozmutlu, and B. J. Jansen. Us versus european web searching trends. *ACM Sigir Forum*, 36(2):32–38, 2002. → *Cited on page 12.*

[121] A. Spink, D. Wolfram, M. B. Jansen, and T. Saracevic. Searching the web: The public and their queries. *Journal of the American society for information science and technology*, 52(3):226–234, 2001. → *Cited on page 25.*

[122] B. Steichen, H. Ashman, and V. Wade. A comparative survey of personalised information retrieval and adaptive hypermedia techniques. *Information Processing & Management*, 48(4):698–724, 2012. → *Cited on pages 4 and 13.*

[123] K. Sugiyama, K. Hatano, and M. Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In *Proceedings of the 13th international conference on World Wide Web*, pages 675–684. ACM, 2004. → *Cited on pages 19 and 26.*

[124] M. Taghavi, A. Patel, N. Schmidt, C. Wills, and Y. Tew. An analysis of web proxy logs with query distribution pattern approach for search engines. *Computer Standards & Interfaces*, 34(1):162–170, 2012. → *Cited on page 76.*

[125] 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):1–34, 2010. → *Cited on pages 14 and 29.*

[126] X. Tao, Y. Li, and N. Zhong. A personalized ontology model for web information gathering. *Knowledge and Data Engineering, IEEE Transactions on*, 23(4):496–511, 2011. → *Cited on page 20.*

[127] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 449–456. ACM, 2005. → *Cited on pages 18, 24, and 26.*

[128] J. Teevan, S. T. Dumais, and E. Horvitz. Potential for personalization. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 17(1):4, 2010. → *Cited on pages 4 and 13.*

[129] J. Trajkova and S. Gauch. Improving ontology-based user profiles. In *RIAO/OAIR: Recherche d'Information Assistée par Ordinateur / Open research Areas in Information Retrieval*, pages 380–390, 2004. → *Cited on pages 18 and 20.*

[130] A. Trotman and B. Sigurbjörnsson. Narrowed Extended XPath I (NEXI). In *Advances in XML Information Retrieval*, pages 16–40. Springer, 2005. → *Cited on pages 44, 63, and 70.*

[131] A. H. Turpin and W. Hersh. Why batch and user evaluations do not give the same results. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 225–231. ACM, 2001. → *Cited on page 28.*

[132] D. Vallet, P. Castells, M. Fernández, P. Mylonas, and Y. Avrithis. Personalized content retrieval in context using ontological knowledge. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(3):336–346, 2007. → *Cited on page 20.*

[133] E. Vicente-López, L. M. de Campos, J. M. Fernández-Luna, and J. F. Huete. Personalization of parliamentary document retrieval using different user profiles. In *Proceedings of the 2nd International Workshop on Personalization in eGovernment Services and Applications (PeGOV'14), in conjunction with the 22nd Conference on User Modeling, Adaptation and Personalization (UMAP'14)*, pages 28–37. Springer, 2014. → *Cited on page 6.*

[134] E. Vicente-López, L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, A. Tagua-Jiménez, and C. Tur-Vigil. An automatic methodology to evaluate personalized information retrieval systems. *User Modeling and User-Adapted Interaction (UMUAI)*, to appear(to appear):1–37, 2014. → *Cited on page 6.*

[135] H. Wang, X. He, M.-W. Chang, Y. Song, R. W. White, and W. Chu. Personalized ranking model adaptation for web search. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 323–332. ACM, 2013. → *Cited on page 24.*

[136] R. W. White. Contextual simulations for information retrieval evaluation. In *Proceedings of the ACM SIGIR 2005 Workshop on Information Retrieval in Context (IRiX)*, page 27. ACM, 2005. → *Cited on page 99.*

[137] R. W. White, I. Ruthven, J. M. Jose, and C. Van Rijsbergen. Evaluating implicit feedback models using searcher simulations. *ACM Transactions on Information Systems (TOIS)*, 23(3):325–361, 2005. → *Cited on pages 31 and 104.*

[138] D. H. Widyantoro, J. Yin, M. El Nasr, L. Yang, A. Zacchi, and J. Yen. Alipes: A swift messenger in cyberspace. In *Proceedings of Spring Symposium Workshop on Intelligent Agents in Cyberspace*, pages 62–67, 1999. → *Cited on page 21.*

[139] I. H. Witten, A. Moffat, and T. C. Bell. *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann, 1999. → *Cited on page 40.*

[140] Y. Yang and B. Padmanabhan. Evaluation of online personalization systems: A survey of evaluation schemes and a knowledge-based approach. *Journal of Electronic Commerce Research*, 6(2):112–122, 2005. → *Cited on page 98.*

[141] D. Zhou, S. Lawless, and V. Wade. Improving search via personalized query expansion using social media. *Information retrieval*, 15(3-4):218–242, 2012. → *Cited on page 23.*

[142] L. Zighelnic and O. Kurland. Query-drift prevention for robust query expansion. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 825–826. ACM, 2008. → *Cited on pages 23 and 26.*

[143] J. Zobel. How reliable are the results of large-scale information retrieval experiments? In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 307–314. ACM, 1998. → *Cited on pages 28 and 78.*