J Intell Inf Syst (2018) 51:597–620
https://doi.org/10.1007/s10844-018-0498-3

Predicting IR personalization performance using pre-retrieval query predictors

Eduardo Vicente-López[1] · Luis M. de Campos[1] · Juan M. Fernández-Luna[1] · Juan F. Huete[1]

Received: 2 March 2017 / Revised: 4 December 2017 / Accepted: 22 January 2018 /
Published online: 30 January 2018

Abstract Although personalization generally improves query performance, it may also occasionally harm how queries perform. If we are able to predict and therefore disable personalization for such situations, overall performance will be higher and users will be more satisfied with personalized systems. We use various state-of-the-art, pre-retrieval query performance predictors and propose several others including user profile information for this purpose. We study the correlations between these predictors and the difference between personalized and original queries. We also use classification and regression techniques to improve the results and finally achieve slightly more than one third of maximum ideal performance. We consider this to be a good starting point within this research line, which will undoubtedly result in further work and improvements.

Keywords Personalization · Information retrieval · Query difficulty · Performance prediction

1 Introduction

There has been an exponential growth in the amount of digital information in recent years and this has made it even more difficult for information retrieval systems (IRSs) to provide truly

✉ Luis M. de Campos
lci@decsai.ugr.es

Eduardo Vicente-López
evicente@decsai.ugr.es

Juan M. Fernández-Luna
jmfluna@decsai.ugr.es

Juan F. Huete
jhg@decsai.ugr.es

[1] Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S.I.I.T., CITIC-UGR, Universidad de Granada, 18071 Granada, Spain

relevant results. Additionally, traditional IRSs retrieve the same list of results for a given query regardless of who submitted it, although the relevance judgments for a given query differ greatly for different users (Teevan et al. 2010). This is known as the *'one-size-fits-all'* problem. Modern IRSs must incorporate new features to tackle these two problems in order to increasingly improve performance while better satisfying specific user information needs.

Personalization (Steichen et al. 2012; Ghorab et al. 2013) is one of these new features. Personalized IRSs consider certain knowledge about the user within the retrieval process in order to suitably adapt the retrieved results to that specific user. Generally speaking, users are more satisfied with personalized IRSs but this is not always the case.

There is a large number of articles (Liu et al. 2004; Dou et al. 2007; Micarelli et al. 2007; Zhou and Yao 2010; de Campos et al. 2014) that show different ways of integrating personalization into different IRSs and their corresponding improvement in performance. However, there is one feature that is not generally mentioned; while personalization generally improves IRS performance (for most queries in fact), it also worsens the performance of other queries (Teevan et al. 2008; Dou et al. 2007). Furthermore, a query could offer good personalized performance results for a given user but not for another, depending on their user profiles. Therefore, the next step would be to be able to predict this performance in order to apply it or not for each individual query and user.

The personalization performance prediction problem has many similarities with classical IR query performance prediction (Carmel et al. 2006), where the quality of the retrieved results for some queries may be poor although the corresponding IRS generally performs well. These poor performance queries are called *difficult* queries, and they should be identified so that they can be properly handled. A difficult query is one that has so many possible answers that it is difficult to choose the correct or the most appropriate one. Since very specific or focused queries are easy to answer, they are not difficult queries. One example of a difficult query, however, is an ambiguous query.

Intuitively, it is easy to see how difficult queries will benefit from personalization as personalization narrows the focus of the user's query and aids disambiguation, among other things. Therefore, the hypothesis is that the prediction of difficult queries and personalization performance will be correlated to some degree.

There are many articles that address the query difficulty prediction problem (Carmel et al. 2006; Hauff et al. 2008b; He et al. 2008; Carmel and Yom-Tov 2010; Shtok et al. 2012; Dan and Davison 2016). These attempt to predict query performance to make use of adaptive retrieval components in order to improve IRS performance. Query difficulty predictors may be classified into two categories: *pre-retrieval* predictors (before the retrieval step), which make use of the information about the query and document collection at indexing time, and *post-retrieval* predictors (after the retrieval step), which make use of the retrieved results and user interaction behavior for the query.

We have only focused on pre-retrieval predictors mainly for two reasons. The first reason is that we want our findings to be usable in real production IRSs and so time response is important. Pre-retrieval predictors can be calculated much faster since they only use information available at indexing time, in contrast to post-retrieval predictors which use different characteristics of the retrieved results (mostly in web environments). The second reason is that our experimental environment is on a much lower scale than usual online IRSs so we do not have enough information about the query background such as the number of times it has been sent, click entropy (variability of results that people click on), etc.

Our final objective is to determine whether to apply personalization for a given query and user profile prior to the retrieval process. We have addressed this personalization performance prediction problem by using a comprehensive set of state-of-the-art, pre-retrieval query difficulty predictors in additional to others that we have proposed. We have correlated their values with those of the difference between the performance of the personalized and the original query.

Although some of these predictors are correlated with personalization performance, the correlations are not very high. For this reason, in order to make the final decision and in order to improve prediction results, we have combined the potential benefit of each individual pre-retrieval predictor using classification and regression techniques. Our results show that we are able to achieve approximately one third of the ideal performance (which would be to be able to identify every query where personalization obtains worse results than the original query, and therefore not apply it to that query and user profile). By disabling personalization for these cases, we obtain slightly better IRS performance than the strategy of applying personalization to every query and user. As far as we are aware, no other article has conducted such a comprehensive study and tested their prediction models to give the improvement in real performance of their predictions over always applying personalization.

The rest of the article is organized as follows: Section 2 reviews related work; Section 3 describes the pre-retrieval predictors used in order to estimate personalization performance and explain why classification and regression techniques have been used with these predictors; Section 4 shows the evaluation environment and the results obtained; and finally Section 5 outlines the general conclusions of the article and proposals for future research.

2 Related work

Our final objective is to predict whether or not to use personalization for a given query and user. As mentioned in the introduction, this problem is related to the problem of identifying *difficult* queries, based on the assumption that these queries will benefit much more from personalization. Accordingly, the articles in the literature use query difficulty predictors to resolve the personalization decision problem. We therefore begin by reviewing articles on query difficulty predictors and finish with those papers that use them for our personalization decision problem.

Query difficulty predictors These are also known as query performance predictors (QPP). Many articles have attempted to find features to serve as indicators to identify query difficulty or performance. The predictors are classified into two different categories: pre-retrieval and post-retrieval.

Pre-retrieval predictors These attempt to predict the query difficulty based on certain characteristics of the query and document collection without using the retrieved results for that query. Since they use some statistics calculated at indexing time they can be computed very fast, which is very important if we want to use them in production environments. They make use of query features, such as the number or average length of query terms, or document collection features such as the inverse document frequency statistics.

Many articles have evaluated the performance of pre-retrieval predictors but each uses a different evaluation framework which makes it impossible to compare their findings. The broadest study in this respect was carried out in Hauff (2010).

These predictors can be classified into *linguistic* and *statistical* methods (Carmel and Yom-Tov 2010). Linguistic approaches use natural language processing and external semantic resources to search for query ambiguity and polysemy. Statistical approaches check deviations in the distribution of query term frequency within the document corpus.

Linguistic approaches exhibit poor query performance prediction as different articles show. In Mothe and Tanguy (2005) the authors extract a set of 16 linguistic features of the query with the help of some linguistic tools, such as syntactical parsers, morphological analyzers and polysemy using WordNet. Most of these linguistic features do not correlate well with system performance. Similar results are presented in Hauff (2010) where semantic distances between query terms are calculated using the WordNet taxonomy.

Statistical predictors are classified in turn into four categories: specificity of the query, similarity between the query and the collection, coherence of the query term distribution, and finally query term relatedness. A comprehensive review of these categories can be found in Carmel and Yom-Tov (2010). We shall now briefly review each of these.

The *specificity* predictors measure query term distribution in the document collection. Therefore, the more specific a query is, the easier it will be to answer and the better its retrieval performance (He and Ounis 2004). They exploit query term statistics such as the *idf* or *ictf* (Zhao et al. 2008). Considering the *similarity* predictors, the greater the similarity between the query and the corpus, the easier the query will be to answer due to the high number of potential relevant documents. In Zhao et al. (2008) the vector-space query similarity to the collection is calculated by considering the entire document corpus as a single large document. An example of the *coherency* approach is presented in He et al. (2008) which measures the inter-similarity of documents containing the query terms. A much less computationally expensive approach is presented in Zhao et al. (2008) by measuring the variance of the weights of the terms in the collection documents that contain them. Each term weight is determined by the IRS weighting scheme, e.g. the widely used *tf-idf*. Finally, we have *term relatedness* predictors, which are based on the assumption that highly related query terms lead to well-formed, easy-to-answer queries (Hauff et al. 2008a).

Post-retrieval predictors These predictors analyze the top ranked results retrieved by a query. Therefore, they are much more complex and time-consuming than their counterpart pre-retrieval predictors. Additionally, since they make use of the IRS list of retrieved results, these predictors depend heavily on the retrieval model being used. However, as they are more complex and use more information than the pre-retrieval predictors (e.g. diversity of results or user interaction with these results), they usually offer a slightly higher prediction quality.

These predictors can be classified into four main categories: *clarity*-based methods, *robustness*-based methods, *score distribution*- based methods, and *user behavior*-based methods.

Clarity-based methods measure the focus or coherence of the list of results in relation to the corpus. For a good performance query, we expect to find a common vocabulary or language between the query and its results. More specifically, if the language of the query results is significantly different from the language of the entire corpus, this means that the query is focused and good performance results are expected. The first definition of clarity

predictors was given in Cronen-Townsend et al. (2002) and is still considered as the state-of-the-art. Some other clarity-based approaches have been proposed such as for example in Amati et al. (2004).

Robustness-based methods measure the robustness of the query results and the greater the robustness of the results, the easier the query is. This can be measured in terms of three different components: firstly, perturbations in the query, i.e. if small changes in the query lead to large changes in the list of results, the query is difficult (Zhou and Croft 2007); secondly, document perturbations, i.e. if the introduction of noise does not significantly affect the ranking of results then the query is robust (Zhou and Croft 2006); and thirdly, perturbations in the retrieval models, i.e. if different retrieval models produce very similar list of results the query is robust and easy to answer (Aslam and Pavlu 2007).

Score distribution-based methods are a less expensive alternative to the two previous approaches, since they do not analyze the top ranked documents but their retrieval score distribution (Zhou and Croft 2007). If low scores are observed in the top ranked results, this is likely due to a difficult query. A more recent approach was introduced in Shtok et al. (2012), where the authors predict query performance by estimating query-drift as the standard deviation of retrieval scores in the top ranked list of results.

User behavior-based methods consider user interactions with the query results to predict query difficulty. In fact, in Guo et al. (2010) the authors use different sources of evidence from queries, results and user interaction logs to train a regression model. Their findings show how user interactions reflect a strong signal of query results quality, as shown by the fact that the top two predictive power features are the average click position of results and average number of clicks.

In Hauff et al. (2010) the authors present a study where they check whether the query performance predictors are correlated with the relevance values assigned by real users. They show that for query suggestions the ratings are mostly uncorrelated, while certain predictors are moderately correlated. These findings suggest that the intuitions behind such predictors are not sufficiently representative of how users rate query results. This calls for further research into proposing new predictors which better capture the user's perception of relevance.

Personalization performance predictions In contrast to the extensively studied research area of query performance predictors, few articles focus on predicting personalization performance. In this case, the objective is clear and a perfect personalization performance predictor (PPP) is the one that is always able to identify those personalized queries that outperform the corresponding non-personalized (original) ones.

Generally speaking, personalization improves the original query performance but it may even harm search accuracy in certain situations (Dou et al. 2007). In this particular article, the authors present a large-scale evaluation framework for personalized search based on query logs. It also demonstrates how click-based personalization techniques perform better than profile-based ones. However, unfortunately this implies having access to incredibly large IRSs logs, which is seldom possible.

The first approach within this specific area is Teevan et al. (2008). The authors extract certain query features, explicit relevance judgments and large-scale log analysis of user behavior for each query to study the variability in user intent, i.e. what each user finds relevant to the same query according to their clicks on the list of results (click entropy), and

attempt to identify queries with the greatest variability or click entropy among users (the ones that most benefit from personalization) as different users find different results relevant. This can also be seen as a measure of query ambiguity. The authors find the click entropy and potential for personalization at 10 (Teevan et al. 2010) as the two most correlated features with implicit measures of query ambiguity. With all this data, they build query ambiguity predictive models to identify queries that can benefit from personalization. However, they do not actually compare the performance of personalized and non-personalized approaches to check whether their model is helpful or not.

Another article with the same problem is Chen et al. (2010), where the authors use classification and regression techniques but in a completely different way to us. They attempt to predict whether certain given predictors correctly predict the variability of the retrieved relevant results for different users. They rely on the assumption that the higher the variability, the higher the personalization performance potential, but they do not compare the results obtained by an original query with those obtained by the personalized one, as we do.

The most similar approach to ours in the literature is Younus et al. (2013). The authors study the correlations between three pre-retrieval and two post-retrieval query difficulty predictors and personalization performance using explicit relevance judgments from 25 real users. One of the post-retrieval predictors presents the best correlation, and they generally find that when standard QPP methods say that a query is difficult, the performance of this query improves with the use of personalization.

Finally, we can also find the personalization prediction problem applied to another domain such as recommender systems (Zhang et al. 2013). The authors analyze the ranking of recommendation lists and from a risk management perspective they provide a technique to predict whether personalization will be helpful. The resulting switching algorithm, which decides whether to apply personalization, outperforms common recommendation algorithms.

3 Pre-retrieval predictors for personalization performance

Despite the a priori inferiority of pre-retrieval predictors with regard to post-retrieval predictors (mainly because they have much less information to make their predictions), various experiments reveal a reasonably good performance which is even comparable to some of the much more complex post-retrieval approaches (Carmel and Yom-Tov 2010). As we mentioned in the introduction, we shall use and focus on the pre-retrieval predictors, mainly because we want our findings to be usable in real production IRSs and because we do not have enough background information.

We shall now show and explain the basics of the comprehensive set of pre-retrieval predictors we have used in this article. It is worth noting that the query terms are stemmed and stopwords are removed before the predictor values are calculated.

The first two approaches are two simple linguistic predictors.

The number of terms in the query (*numQT*). This predictor is based on the assumption that the higher the number of terms in the query, the more specific and better explained the query will be.

$$numQT = \sum_{t \in Q} 1, \tag{1}$$

where Q is the query and t is each query term.

The average query term length (*avgQTL*). This predictor is based on the assumption that longer terms are less common in the corpus and so are more specific.

$$avgQTL = \left(\sum_{t \in Q} t_l \right) / numQT, \tag{2}$$

where t_l is the number of characters of term t.

The remaining predictors are based on the collection statistics calculated at the time of indexing and most of these were first seen in He and Ounis (2004) and Zhao et al. (2008).

- The first set of predictors is based on the well-known IR concept of inverse document frequency (*idf*). Each document collection term has its own *idf* value. If the *idf* value is high, this means that the term rarely occurs in documents and so is very specific or selective. The *idf*-based predictors are as follows:

$$sumIDF = \sum_{t \in Q} \log \frac{N}{f_t}, \qquad avgIDF = \frac{sumIDF}{|Q|_{t \in V}}, \qquad maxIDF = max_{t \in Q} \log \frac{N}{f_t}, \tag{3}$$

  where N is the total number of documents in the collection, f_t is the number of documents containing term t, Q is the query, V is the collection vocabulary (unique terms), and $|Q|_{t \in V}$ is the query length for terms in V.

  sumIDF will be biased toward longer queries. Therefore, we normalize *sumIDF* by the query length and only consider those terms in V as *avgIDF*. An alternative normalization approach is to choose the term with the maximum IDF score, i.e. *maxIDF*.
- The second set of predictors is based on another well-known IR concept: inverse collection term frequency (*ictf*). Each document collection term has its own *ictf* value. The assumption is the same as that of *idf* in that high values mean terms are very specific or selective. The *ictf*-based predictors are as follows:

$$sumICTF = \sum_{t \in Q} \log \frac{|C|}{f_{c,t}}, \qquad avgICTF = \frac{sumICTF}{|Q|_{t \in V}}, \qquad maxICTF = max_{t \in Q} \log \frac{|C|}{f_{c,t}}, \tag{4}$$

  where $|C|$ is the number of all the terms in collection C, $f_{c,t}$ is the frequency of term t in C.

  The *SCS* (simplified clarity score) predictor (He and Ounis 2004) is based on the post-retrieval *CS* (clarity score) predictor. *SCS* is strongly related to *avgICTF* and assumes that each term appears only once in the query (a reasonable assumption for all but extra large queries, which are not very frequent). In this case, the *SCS* is calculated as follows:

$$SCS = \log \frac{1}{numQT} + avgICTF. \tag{5}$$

  The predictors in (3), (4) and (5) are classified under the *specificity* category of pre-retrieval predictors, i.e. how specific the query is. The assumption is that queries with low values (sum, average or maximum), which are queries with very frequent terms, are difficult to satisfy. In particular, the *SCS* predictor measures the specificity of the query by also considering the query length.
- The third set of predictors (*SCQ*) are classified under the *similarity* category of pre-retrieval predictors, i.e. how similar are the query and the collection. The assumption

is that queries with low similarity values will be difficult to answer. The *SCQ*-based predictors are as follows:

$$sumSCQ = \sum_{t \in Q} (1 + \ln(f_{c,t})) \ln(1 + N/f_t), \qquad avgSCQ = \frac{sumSCQ}{|Q|_{t \in V}},$$
$$maxSCQ = max_{t \in Q} \left((1 + \ln(f_{c,t})) \ln(1 + N/f_t) \right), \tag{6}$$

where each component has already been explained in the previous equations.

- The fourth set of predictors (*VAR*) are classified under the *coherency* category of pre-retrieval predictors, i.e. they measure the inter-similarity of documents containing the query terms. More specifically, *VAR(t)* measures the variance of the weights of the term t on the documents in the collection containing it. The term weight depends on the retrieval model used. The assumption is that if the variance of the term distribution on the documents containing t is low, the query will be more difficult to answer.

  Each query term t will have a weight value $w_{d,t}$ if it is present in document d. The distribution of t in all collection documents containing it can then be estimated. We use a simple *tf-idf* approach to calculate $w_{d,t}$:

$$w_{d,t} = (1 + \ln(f_{d,t})) \ln(1 + N/f_t),$$

  where $f_{d,t}$ is the frequency of t in document d, with $w_{d,t} = 0$ for query terms not in the collection vocabulary V.

  In order to calculate the variance or dispersion we also need the average weight of term t over the collection ($\overline{w}_t$):

$$\overline{w}_t = \frac{\sum_{d \in D_t} w_{d,t}}{f_t},$$

  where D_t is the set of collection documents containing term t and f_t is its size.

  The *VAR*-based predictors are calculated as follows:

$$sumVAR = \sum_{t \in Q} \sqrt{\frac{1}{f_t} \sum_{d \in D_t} (w_{d,t} - \overline{w}_t)^2}, \qquad avgVAR = \frac{sumVAR}{|Q|_{t \in V}},$$
$$maxVAR = max_{t \in Q} \sqrt{\frac{1}{f_t} \sum_{d \in D_t} (w_{d,t} - \overline{w}_t)^2}. \tag{7}$$

  We do not use any pre-retrieval predictor from the *term relatedness* category, e.g. *PMI*, since our retrieval model does not consider term proximity in its retrieval process. As mentioned in Carmel and Yom-Tov (2010), the queries with strongly related terms will probably be best served by retrieval models which use this term proximity. In Hauff (2010) *maxVAR* and *maxSCQ* are shown to outperform the other predictors. Based on this last statement and because the *maxVAR* and *maxSCQ* assumptions are based on different sources of evidence, we also consider it worth combining both predictors to see whether this improves prediction accuracy.

  We use the same simple interpolation given in Zhao et al. (2008) to combine both kinds of predictors as follows:

$$joint = \alpha\ maxSCQ + (1 - \alpha) sumVAR,$$
$$joint2 = \alpha\ maxSCQ + (1 - \alpha) maxVAR, \tag{8}$$

where α is a parameter to determine the importance of each component within the interpolation. We have selected a value of $\alpha = 0.75$, which is within the maximum performance range as the authors of the previous article suggest. Surprisingly, although the best performance predictors according to Hauff (2010) are *maxVAR* and *maxSCQ*, the authors in Zhao et al. (2008) use *maxSCQ* and *sumVAR* in their *joint* predictor. Therefore, we have also proposed a *joint2* predictor using *maxVAR* instead of *sumVAR*.

3.1 Including the profile in predictors

All of the previous seventeen pre-retrieval predictors only use information from the query and the collection. This is because all of them have been designed as predictors for the query difficulty but not specifically for personalization prediction purposes. When we consider personalization, a third component comes into play, the user profile. There are different ways to collect and represent user information (Vicente-López et al. 2016). In this article, we shall work with user profiles which are represented as a set of weighted keywords. This means that a profile is composed of a set of terms which are considered as highly representative of the user interests and preferences. Each term is associated with a numerical weight expressing the degree of importance of the term within the profile. In our case the term weighting scheme considered is the so-called *diffFreq*, which essentially computes the normalized frequency of the term t within the profile $Prof$ minus the normalized frequency of t outside $Prof$ (see Vicente-López et al. 2016 for more details).

When we apply personalization, if the query terms are very different from the user profile terms, user profile information will steer the query results to user interests and preferences. However, if the user profile terms are very similar to those in the query, the effect of including user profile information will not greatly affect the original query results since the original query was already informative enough and close to the user. Therefore, the greater the difference between the query and the user profile, the higher the impact of personalization. In order to measure this difference (and as our first proposed predictor to consider the user profile) we propose the use of a simple cosine similarity measure between the query and user profile.

The *cosine* predictor is calculated as follows:

$$cosineQP = \frac{\sum_{t \in Q \cap Prof} wq_t \; wp_t}{\sqrt{\sum_{t \in Q} wq_t^2} \sqrt{\sum_{t \in Prof} wp_t^2}}, \tag{9}$$

where wq_t and wp_t are the weights of term t in query Q and profile $Prof$, respectively.

The values of *cosineQP* range from 0 (meaning totally different terms) to 1 (meaning exactly the same terms between the query and the user profile).

The next step is to simultaneously consider the three components involved in the personalization process: the query, the user profile and the document collection. One of the most common ways to personalize a query is to simply add a given number of the user profile terms as expansion terms to the original query. In this case, all of the previously used pre-retrieval predictors that only consider the query and the document collection could be reused, but in this case using the expanded query instead of the original query.

We can therefore modify all of the previous pre-retrieval predictors (excluding *numQT* and *avgQTL*, where the modification is meaningless) in (3) to (8), which will be denoted by adding *QP* to their names, e.g. *sumIDFQP, avgICTFQP* or *maxSCQQP*.

Another approach for taking into account both the query and the profile is to consider two separate queries, the original and the expanded query including the profile terms. We can

compute the difference between *avgIDFQP* and *avgIDF*. If this difference is positive, this means that the expanded query is more specific than the original query and so it is probably easier to satisfy. It should be noted that this argument is valid if we use the *avg* version of the predictor but not for the *sum* version. We can extend this reasoning by analogy to other predictor families to propose the following four predictors:

$$profIDF = avgIDFQP - avgIDF, \qquad profICTF = avgICTFQP - avgICTF,$$
$$profSCQ = avgSCQQP - avgSCQ, \qquad profVAR = avgVARQP - avgVAR. \tag{10}$$

All of the previously proposed approaches make a total of 37 different pre-retrieval PPP and these constitute a fairly comprehensive and heterogeneous set of predictors that focus on different components such as the query, document collection or user profile, and on different aspects such as the specificity, similarity or coherency of these components.

4 Experimental environment and results

This section shows all of the necessary components to perform the evaluation process to check whether we can accurately predict the personalization performance and use these predictions to decide whether to personalize a given query. It also shows the obtained results and our main conclusions.

4.1 Experimental framework

Our document collection comprises a set of official documents from the Andalusian Parliament in XML format. More specifically, it consists of 658 committee sessions from the sixth and seventh terms of office (containing 432,575 retrievable structural units). Each committee session covers a different area of interest such as agriculture, education, economy, etc. Each document contains the transcriptions of the speeches of the Members of Parliament discussing a proposed initiative relating to a given issue in the corresponding committee session.

We use Garnata (de Campos et al. 2006) as the search engine and this is based on probabilistic graphical models. This structured IRS has been tested and improved at three editions of the INEX workshop (de Campos et al. 2009). From the thirteen personalization strategies proposed in (de Campos et al. 2014) we use the so-called HardReranking (HRR) approach. Although this is not the best-performing approach, it does perform best in techniques that are more easily implementable by other IRSs.

For the evaluation of personalized results we still need a set of queries, users and their user profiles and relevance assessments. We have two different sets of such components: a first set from a conducted user study (de Campos et al. 2014) and a second set from an automatic strategy for personalized IRSs evaluation called ASPIRE (Vicente-López et al. 2015).

User study We have a heterogeneous set of 23 queries formulated by real users of the document collection, which represents a small but trustworthy sample of real user information needs. The user study involved 31 users. Each user submitted one or several of the previous 23 queries to the IRS assuming, among a fixed set of generic profiles, the profile(s) that best fits them. These generic user profiles were automatically learned from the content of the documents in each committee session and were represented as sets of weighted terms (Vicente-López et al. 2016). There are eight different generic user profiles relating

to administration, agriculture, culture, economy, education, employment, environment and health. When a user evaluates a query under a given profile, a set of relevance assessments is obtained for this user, profile and query. We call the previous set of relevance assessments an *evaluation triplet*. A total number of 126 evaluation triplets were obtained in the user study. Further information about this user study can be found in de Campos et al. (2014).

ASPIRE The main problem of user studies is the limited number of evaluation triplets obtained due to the enormous effort required to conduct any user study. This problem is solved by ASPIRE. Using this automatic strategy to evaluate personalized IRSs we can automatically generate personalized relevance assessments for any given query and profile. It basically considers a query result as relevant if it belongs to the area of interest the user profile represents and is within the top ranked results, given by a threshold which in our case is equal to 100. Using ASPIRE we have been able to increase the number of queries in order to get more reliable evaluation results. More specifically, for each initiative in the document collection we have used the text within the tag *abstract*, which briefly summarizes the initiative content, as a query to the system. In this way, with ASPIRE we have a total of 2602 queries and evaluation triplets with their corresponding sets of relevance assessments.

We will use normalized discounted cumulative gain (NDCG) (Järvelin and Kekäläinen 2002) as the evaluation metric to measure the IRS retrieval performance for a given list of results. This evaluation metric estimates the cumulative relevance gain observed by a user for the top documents in a retrieved list of results. We specifically evaluate this list of results for the top fifty elements.

We shall now define the improvement in personalization in terms of the original query as the difference between the performances (measured by NDCG) of the personalized and the original query:

$$diffPerso = NDCG@50_{HRR} - NDCG@50_{Orig}. \tag{11}$$

Table 1 shows the distribution of the evaluation triplets of both the user study and ASPIRE approaches for each user profile. It also shows the number of evaluation triplets where *diffPerso* is positive (personalization outperforms the original query), negative (personalization worsens the original query) and equal to zero (there is neither improvement nor loss in performance).

The total number of ASPIRE evaluation triplets is 20.65 times the number of user study triplets. In terms of *diffPerso* distribution for both approaches there are always more positive than negative cases. For a general idea of this distribution, in the user study the ratio of negative cases to positive cases ranges approximately from 10 to 60% with an average of 29%. In ASPIRE this ratio ranges from 6 to 33% (with the exception of the *economy* profile with 75%) and an average of 21% considering all the profiles. The number of *0s* is negligible for both approaches. With this data we may consider both approaches to be roughly comparable in terms of *diffPerso* distribution, although not in terms of the total number of evaluation triplets. However, this was precisely the objective of using ASPIRE, i.e. to increase the number of evaluation triplets in order to obtain more robust results and conclusions.

4.2 Correlations between predictors and *diffPerso*

It should be remembered that our final objective is to discern whether to apply personalization for a given query and user profile before the search is performed. If search performance is higher than the original query performance when personalization is applied, then the user will be more satisfied with the personalized results. The measure of this better or worse user satisfaction can be approximated by the difference in performance between the personalized

Table 1 Number of evaluation triplets according to profile and *diffPerso* for the user study and ASPIRE

	diffPerso	Administration	Agriculture	Culture	Economy	Education	Employment	Environment	Health
User study	+	10	18	14	10	12	10	8	14
	−	4	2	5	6	2	3	4	2
	0s	0	0	0	1	0	1	0	0
	Total	14	20	19	17	14	14	12	16
ASPIRE	+	292	327	376	137	351	169	231	240
	−	76	50	127	103	34	10	18	29
	0s	2	7	4	3	7	0	9	0
	Total	370	384	507	243	392	179	258	269

and the original query (*diffPerso*). If any of the PPP values in Section 3 highly correlates with *diffPerso*, we could use this given PPP to predict if personalization will benefit or harm user-perceived IRS quality. In this way, we will be able to choose whether to activate personalization before the search is performed, thereby maximizing overall IRS performance for the given user.

The correlation tables for the user study and ASPIRE using all of the 37 proposed predictors and 8 user profiles are presented in Appendix A, for reasons of space (two tables, each almost a page long). However, in order to summarize this information, Tables 2 and 3 present the *diffPerso*-predictors average (μ) and maximum (max) correlations grouped according to profiles for the user study and ASPIRE, respectively. We have used the Pearson method implemented in the R statistical framework[1] for every correlation.[2]

We can draw the following conclusions from Tables 2 and 3: firstly, no single predictor has a high enough average correlation to be considered as a good and robust predictor for personalization performance; secondly, average (and particularly maximum) correlations are higher in the user study than with ASPIRE, although this may be due to the low number of evaluation triplets for each profile considered in the user study; and thirdly, *cosineQP* is the best predictor in terms of correlation both on average and maximum values.

According to the previous results, there is a high variability in the correlation values between *diffPerso* and the predictors. These correlations are very dependent on the given predictor and the applied user profile (see Appendix A). Therefore, although some predictors are good for certain profiles, they are not for others. In many cases, even the same predictor gives negative correlation with some profiles and positive with others.

Table 4 shows the ten best predictors based on their average performance across the different user profiles in our experiments. These would be the predictors we would select to predict the personalization performance gain if we were to stop our research here.

As mentioned previously, *cosineQP* is consistently the best predictor, followed by different *SCQ* variations. However, none of the predictors may be considered as a good personalization performance predictor, thereby at least partially canceling the initial hypothesis that prediction of difficult queries and personalization performance are highly correlated. Due to this correlation variability and the fact that each predictor measures different aspects, we think it would be interesting to combine the potential benefit of each individual PPP using classification and regression techniques. In fact, these techniques offer an additional advantage in that they inform us directly of whether to apply personalization, unlike any individual predictor for which we would need to select a threshold for such a decision problem.

4.3 Using classification and regression techniques

In order to improve prediction performance by harnessing the potential of all the predictors by using classification and regression techniques, we have used the WEKA[3] machine learning framework. As correlations highly depend on the user profile applied to the query and these profiles have a different number of evaluation triplets (see Table 1), we have decided to build a different predictive model for each profile. The input data are the values of all

[1] https://cran.r-project.org/

[2] We have also used the Spearman and Kendall methods to compute the correlations and similar results were obtained.

[3] http://www.cs.waikato.ac.nz/ml/weka/

Table 2 *diffPerso*-predictors average (μ) and maximum (max) correlation values grouped by profiles for the user study

Predictor	μ	max	Predictor	μ	max	Predictor	μ	max	Predictor	μ	max
numQT	− 0.125	− 0.572	avgSCQ	− 0.066	− 0.463	maxIDFQP	− 0.101	− 0.372	maxVARQP	− 0.042	− 0.471
avgQTL	0.023	0.41	maxSCQ	− 0.232	− 0.751	sumICTFQP	− 0.085	− 0.619	jointQP	− 0.062	− 0.492
sumIDF	− 0.204	− 0.725	sumVAR	− 0.164	− 0.479	avgICTFQP	− 0.052	− 0.542	joint2QP	− 0.042	− 0.471
avgIDF	− 0.048	− 0.472	avgVAR	− 0.193	− 0.491	maxICTFQP	− 0.192	− 0.535	profIDF	0.041	0.459
maxIDF	− 0.216	− 0.673	maxVAR	− 0.174	− 0.481	SCSQP	− 0.028	− 0.494	profICTF	− 0.092	− 0.57
sumICTF	− 0.085	− 0.619	joint	− 0.164	− 0.489	sumSCQQP	− 0.223	− 0.755	profSCQ	0.06	0.451
avgICTF	0.085	0.572	joint2	− 0.173	− 0.493	avgSCQQP	− 0.219	− 0.681	profVAR	0.149	0.482
maxICTF	− 0.126	− 0.483	cosineQP	− 0.254	− 0.755	maxSCQQP	− 0.027	− 0.052	–	–	–
SCS	0.14	0.583	sumIDFQP	− 0.204	− 0.725	sumVARQP	− 0.062	− 0.492	–	–	–
sumSCQ	− 0.223	− 0.755	avgIDFQP	− 0.193	− 0.652	avgVARQP	− 0.038	− 0.509	–	–	–

Table 3 *diffPerso*-predictors average (μ) and maximum (max) correlation values grouped by profiles for ASPIRE

Predictor	μ	max	Predictor	μ	max	Predictor	μ	max	Predictor	μ	max
numQT	− 0.016	− 0.315	avgSCQ	− 0.105	− 0.206	maxIDFQP	0.072	0.122	maxVARQP	0.026	0.137
avgQTL	− 0.093	− 0.203	maxSCQ	− 0.012	− 0.142	sumICTFQP	− 0.002	− 0.229	jointQP	0.016	0.187
sumIDF	− 0.002	0.164	sumVAR	0.024	0.137	avgICTFQP	0.042	− 0.189	joint2QP	0.026	0.137
avgIDF	− 0.028	0.184	avgVAR	0.063	0.141	maxICTFQP	0.081	0.158	profIDF	0.032	− 0.185
maxIDF	0.054	0.108	maxVAR	0.03	0.137	SCSQP	0.041	0.288	profICTF	− 0.003	0.184
sumICTF	− 0.002	− 0.229	joint	0.024	0.137	sumSCQQP	− 0.031	− 0.304	profSCQ	0.11	0.203
avgICTF	0.008	− 0.186	joint2	0.03	0.137	avgSCQQP	− 0.043	− 0.201	profVAR	− 0.068	− 0.132
maxICTF	0.074	0.141	cosineQP	− 0.279	− 0.376	maxSCQQP	0.029	0.059	–	–	–
SCS	− 0.018	0.292	sumIDFQP	− 0.002	0.164	sumVARQP	0.016	0.187	–	–	–
sumSCQ	− 0.031	− 0.304	avgIDFQP	0.017	− 0.173	avgVARQP	0.03	0.185	–	–	–

Table 4 Ten best *diffPerso*-predictors average correlation values for the user study and ASPIRE

User study		ASPIRE	
Predictor	μ	Predictor	μ
cosineQP	− 0.254	cosineQP	− 0.279
maxSCQ	− 0.232	profSCQ	0.11
sumSCQ	− 0.223	avgSCQ	− 0.105
avgSCQQP	− 0.219	avgQTL	− 0.093
maxIDF	− 0.216	maxICTFQP	0.081
sumIDF	− 0.204	maxICTF	0.074
avgIDFQP	− 0.193	maxIDFQP	0.072
avgVAR	− 0.193	profVAR	− 0.068
maxICTFQP	− 0.192	avgVAR	0.063
maxVAR	− 0.174	maxIDF	0.054

the predictors (which play the role of feature variables) for each evaluation triplet. The last variable (the class) is *diffPerso*. As *diffPerso* is a numeric value it is used directly for regression but must be categorized for classification. Our prediction problem is a binary decision: whether to personalize (*'yes'*) or not (*"no"*). We have categorized *diffPerso* in the following way: if *diffPerso* < 0, we transform it into the nominal *"no"* (personalization worsens the original query performance), if $diffPerso > 0$, we transform it into the nominal *"yes"* (personalization improves the original query performance), and if *diffPerso* $= 0$, we delete this observation, since it does not provide any useful information on personalization (it may be considered as 'noise').

We have tried several classification and regression algorithms implemented in WEKA, but finally we have decided to use the *Random Forest* approach, since it provides the best results and is suitable for both classification and regression.

Personalization almost always outperforms the original query performance. For this reason, the approach followed by almost every personalized IRS is to always personalize all of the queries. This affirmation is validated by our own results, representing 76 and 82 percent of all the triplets (see Table 1) for the user study and ASPIRE, respectively. Roughly speaking, we could say that 3 out of 4 queries are helped by personalization. For this reason, the baseline approach that we use for comparison purposes is always to personalize.

We have applied a *leave-one-out (LOO)* approach for the user study since it has only a few observations for each profile (see Table 1). For ASPIRE we have used a 10-fold cross validation approach to evaluate the prediction model.

Tables 5 and 6 show the results of the learned models. *avgPerso* is the NDCG average on all the test observations by always using (*baseline*) personalization. The next two columns *avgPred* and *% gain* are the averages over all the test observations and the percentage gain against *avgPerso* using the best performance approach, i.e. this *ideal* value would be obtained if we were able to always make the best decision about using the original or personalized query. The next four columns are the results following the predictions given by the built classification and regression models. The last row in the tables, "μ *IdealGain %*", represents the percentage of the ideal *% gain* we are able to obtain by following our prediction models.

Table 5 Leave-one-out random forest classification and regression prediction results for the user study

	(baseline)	Ideal		Classification		Regression	
User profile	avgPerso	avgPred	% gain	avgPred	% gain	avgPred	% gain
Administration	0.608051	0.644634	6.01644	0.570279	− 6.21198	0.592527	− 2.55308
Agriculture	0.79087	0.810896	2.53215	0.774176	− 2.11084	0.774176	− 2.11084
Culture	0.638095	0.718654	12.62492	0.689769	8.09817	0.689769	8.09817
Economy	0.40522	0.545995	34.74039	0.474091	16.99595	0.421873	4.10962
Education	0.563201	0.592569	5.21448	0.563201	0	0.563201	0
Employment	0.617909	0.656129	6.18538	0.614254	− 0.59151	0.627099	1.48727
Environment	0.611982	0.687295	12.30641	0.641862	4.8825	0.658471	7.59647
Health	0.717045	0.722	0.7132	0.717045	0	0.717045	0
μ	0.619047	0.672291	10.04167	0.630585	2.63279	0.63052	2.07845
μ *IdealGain %*	–	–	–	–	26.22	–	20.70

We can draw the following conclusions from Tables 5 and 6. Firstly, all *% gain* (ideal, classification and regression) are less robust between profiles in the user study than in ASPIRE. Secondly, the ideal *% gain* is relatively low for both approaches, and the value for the user study is considerably higher than for ASPIRE, probably because of the lower robustness of the user study. Thirdly, for the user study the classification and regression *% gains* are similar on average but different across profiles. For the user study in the best case (classification) we are able to reach 26% of the ideal performance. Fourthly, for ASPIRE the *% gain* obtained by the regression model clearly outperforms the value obtained by the classification model. In this case we are able to obtain 39% of the ideal performance.

We have greater confidence in the ASPIRE results because in the user study there are not enough data to obtain robust results, even when the LOO approach is used to build the predictive models. Another sign for this lack of robustness is that in this problem (even

Table 6 Random forest classification and regression prediction results for ASPIRE

	(baseline)	Ideal		Classification		Regression	
User profile	avgPerso	avgPred	% gain	avgPred	% gain	avgPred	% gain
Administration	0.703849	0.736331	4.61491	0.719888	2.27876	0.716649	1.81857
Agriculture	0.834488	0.858497	2.87709	0.823421	− 1.3262	0.846435	1.43166
Culture	0.782663	0.80918	3.38805	0.793442	1.37722	0.794417	1.5018
Economy	0.688553	0.724375	5.2025	0.699786	1.63139	0.705563	2.4704
Education	0.841959	0.844191	0.2651	0.84129	− 0.07946	0.842851	0.10594
Employment	0.779812	0.78025	0.05617	0.779812	0	0.779812	0
Environment	0.801694	0.804977	0.40951	0.80154	− 0.01921	0.80154	− 0.01921
Health	0.818913	0.82685	0.96921	0.811543	− 0.89997	0.816464	− 0.29905
μ	0.781491	0.798081	2.22282	0.783840	0.37032	0.787966	0.87626
μ *IdealGain %*	–	–	–	–	16.66	–	39.42

Table 7 Confusion matrices for the *education* and *administration* profiles obtained by the regression model with ASPIRE

	Education		Administration	
	Predicted		Predicted	
	Yes	No	Yes	No
Personalize yes	72	0	49	5
Personalize no	2	2	12	8

though it is a better fit for regression since the class is numeric and some of the information is lost through categorization) classification performs better under the user study, unlike ASPIRE. We show the user study results for comparison purposes with ASPIRE and mainly because their relevance assessments were provided by real users. However, unfortunately they do not represent enough data to enable generalizations to be made on the basis of them. Other things we have tried in order to improve the ASPIRE results is to resample the observations by following different strategies due to the imbalanced dataset particularly in some profiles. Another attempt was to learn only one general classifier and regressor for each profile. Both attempts gave us worse results than those in Table 6.

Notice that we are not measuring directly the accuracy of the classifiers (or the regressors), i.e. we are not computing the number of true and false positives and the number of true and false negatives. The reason is that these quantities are not the important ones. What it is really important is how the decisions made by the classifiers/regressors affect the performance of the complete system, i.e. whether the user receives a better ranking of documents (measured through the NDCG metric). For example, it is possible that the classifiers make some mistakes (either wrongly deciding to personalize a query that should not be personalized or deciding not to personalize a query that should be personalized) but the difference in performance between personalizing or not these queries is very low, whereas the classifier gets right on another query (correctly deciding to personalize or not this query) and in this case the difference is very high. In this situation, the confusion matrix could give us the wrong impression that the system performs badly when in fact it is improving the global NDCG values. Anyway, we include in Table 7 examples of confusion matrices for two profiles, in order to give the reader a more complete picture of the situation.

In Table 7 we can observe the confusion matrices for two of the profiles, *education* and *administration*, obtained by the regression model with ASPIRE. In both cases the number of true positives is considerably higher than the number of true negatives,[4] and also higher than the number of false positives and false negatives. The accuracy is quite good, 97% for *education* and 77% for *administration*. However, notice that although the number of errors is considerably higher in the case of *administration* (17 errors out of 74) than in the case of *education* (2 errors out of 76), the percentage of NDCG gain (see Table 6) in the first case, 1.82%, is much higher than the percentage in the second case, 0.11% (although in both cases they represent around 39% of the corresponding ideal gain).

The problem (depending on how you look at it) is that applying personalization benefits almost every query. It is therefore very difficult to accurately predict which queries do not benefit without wrongly identifying those that do. There is a low maximum gain value from which we are able to reach slightly more than one third.

[4] so that the tendency of the system is to personalize most of the queries.

Table 8 Random forest classification and regression prediction results for ASPIRE using the 10 highest correlation predictors from Table 4

	(baseline)	Ideal		Classification		Regression	
User profile	avgPerso	avgPred	% gain	avgPred	% gain	avgPred	% gain
Administration	0.703849	0.736331	4.61491	0.716764	1.83491	0.722564	2.65895
Agriculture	0.834488	0.858497	2.87709	0.829471	− 0.60121	0.829797	− 0.56214
Culture	0.782663	0.80918	3.38805	0.788394	0.73224	0.789665	0.89464
Economy	0.688553	0.724375	5.2025	0.698906	1.50359	0.710254	3.15168
Education	0.841959	0.844191	0.2651	0.84129	− 0.07946	0.842145	0.02209
Employment	0.779812	0.78025	0.05617	0.779812	0	0.779812	0
Environment	0.801694	0.804977	0.40951	0.80154	− 0.01921	0.80154	− 0.01921
Health	0.818913	0.82685	0.96921	0.814721	− 0.5119	0.818077	− 0.10209
μ	0.781491	0.798081	2.22282	0.783862	0.35737	0.786732	0.75549
μ *IdealGain %*	–	–	–	–	16.08	–	33.99

Predictors selection In order to obtain the previous prediction performance results we need to calculate all of the 37 pre-retrieval predictors proposed in this article. The next step is to check if we are able to reach similar prediction values using a considerably lower number of predictors, which obviously will be faster in calculation and response time. We will do this process only for ASPIRE because its results are more robust and trustworthy.

For this task, we could follow two different alternatives: either to use any of the automatic feature selection strategies available in Weka or to manually select the features with the highest correlations from Table 4. We have explored some of the strategies from the first alternative and these include *CfsSubsetEval*, which provides a set of predictors by considering the individual predictive ability of each predictor along with the degree of redundancy between them; *CorrelationAttributeEval* and *InfoGainAttributeEval*, which provide a ranked list of predictors by measuring the correlation and the information gain, respectively, between them and the class, and others with different parameter configurations with no good final results. However, if we use the 10 predictors with the highest ASPIRE correlations from Table 4, then we obtain almost the same prediction performance results as when all the proposed predictors are used. Table 8 shows these new results.

Looking at the results of Table 8, we can see how almost exactly the same prediction performance is achieved by using classification and a slightly worse performance in the case of regression, although it is still possible to capture 1 out of 3 triplets where personalization harms the original query performance. If we only use the five predictors with the highest correlation values from Table 4, the prediction power drops to approximately half that of when all the predictors are used and this is not acceptable.

We can therefore conclude that if the final IRS time response is critical then only the ten ASPIRE predictors from Table 4 should be used to decide whether to personalize the user query since almost the same prediction performance is reached when all the predictors proposed in this article are used. If the IRS time response is not so critical, both approaches could be used, since their difference in time computation is not significant against the time required to perform the search.

5 Conclusions and future work

In this article we have tackled the difficult task of predicting whether personalization will benefit or harm the original query performance before the search is performed. If we are able to identify the harmed queries, the personalization module could be deactivated for those cases in order to obtain the maximum performance from the personalized IRS. Most of the time personalization outperforms the original query performance but this is not always true and it will depend both on the query and the user.

In the literature this personalization prediction problem has been related to the problem of predicting difficult queries. A difficult query is one that has so many possible answers that it is difficult to retrieve the most appropriate ones. This normally happens when the query is very short, ambiguous or its topic is very general. In such situations, personalization helps provide results which are closer to the user, who will in turn be more satisfied with the IRS.

We have performed a comprehensive study using most of the state-of-the-art, pre-retrieval query difficulty predictors. These predictors are based on different assumptions of how users assign relevance to the list of retrieved results. Since both the query and profile affects personalization results, we have extended the previous predictors and proposed others to also include the profile information in the personalization prediction problem. We have finally used a comprehensive and heterogeneous set of 37 pre-retrieval predictors.

We have correlated these predictors with *diffPerso*, the difference between the performance of the personalized query and the original query. Since these correlations are not very high, there is no single predictor that could be considered good enough to predict personalization performance. Consequently, we have attempted to obtain the most out of each predictor potential, based on different assumptions, by considering all of them together through the use of classification and regression techniques.

As far as we aware, nobody else has conducted such a comprehensive study, including the use of machine learning techniques, and given the final improvement of their personalization prediction models against the logical baseline of always applying personalization to every query. The personalization prediction models we have built are able to improve personalization performance and obtain slightly more than one third of the maximum reachable ideal performance, i.e. to be able to identify and therefore disable personalization for all of the personalized queries with lower performance than their corresponding original queries. We also finally prove that by only using the 10 predictors with the highest *diffPerso* correlations (and not each of 37 proposed in the article) almost the same improvement could be reached. This may be important for IRSs where query response time is critical.

We generally believe that the results discussed in this article are promising and a good starting point for further research in this area. We think new predictors are needed, some of which would probably use new ways to include the user profile information, to improve our results and be as close as possible to the maximum ideal personalization performance. In addition to proposing new personalization performance predictors, another line of future work could be to not include all the user profile information but only that which is most relevant to the given query, and this is particularly important if the profiles are heterogeneous and represent several areas of interest.

Acknowledgements This work has been supported by the Spanish Andalusian "Consejería de Innovación, Ciencia y Empresa" postdoctoral phase of project P09-TIC-4526, the Spanish "Ministerio de Economía y Competitividad" projects TIN2013-42741-P and TIN2016-77902-C3-2-P, and the European Regional Development Fund (ERDF-FEDER).

Appendix A

Table 9 *diffPerso*-predictors correlation values by user profile and predictor for the user study

	Administration	Agriculture	Culture	Economy	Education	Employment	Environment	Health
numQT	0.036	− 0.201	− 0.035	0.005	0.087	− 0.308	− 0.015	− 0.572
avgQTL	0.41	− 0.211	− 0.006	0.059	0.237	− 0.027	0.053	− 0.334
sumIDF	− 0.193	− 0.104	− 0.223	0.396	− 0.245	− 0.068	− 0.474	− 0.725
avgIDF	− 0.008	0.064	− 0.218	0.45	− 0.169	0.368	− 0.401	− 0.472
maxIDF	− 0.299	0	− 0.446	0.391	− 0.229	0.082	− 0.554	− 0.673
sumICTF	0.02	− 0.01	− 0.052	0.418	− 0.162	− 0.011	− 0.263	− 0.619
avgICTF	0.14	0.25	0.008	0.572	− 0.198	0.526	− 0.299	− 0.316
maxICTF	− 0.237	0.15	− 0.471	0.464	− 0.242	0.237	− 0.428	− 0.483
SCS	0.15	0.315	0.068	0.501	− 0.197	0.583	− 0.197	− 0.105
sumSCQ	− 0.134	− 0.269	− 0.135	0.144	− 0.032	− 0.261	− 0.34	− 0.755
avgSCQ	− 0.061	− 0.044	− 0.146	0.355	− 0.099	0.359	− 0.463	− 0.433
maxSCQ	− 0.35	− 0.101	− 0.345	0.313	− 0.12	0.148	− 0.751	− 0.649
sumVAR	− 0.174	− 0.088	− 0.151	0.215	− 0.301	− 0.248	− 0.087	− 0.479
avgVAR	− 0.184	− 0.092	− 0.277	0.161	− 0.296	− 0.258	− 0.111	− 0.491
maxVAR	− 0.179	− 0.084	− 0.18	0.17	− 0.304	− 0.25	− 0.082	− 0.481
joint	− 0.182	− 0.09	− 0.158	0.254	− 0.301	− 0.243	− 0.101	− 0.489
joint2	− 0.187	− 0.087	− 0.188	0.215	− 0.303	− 0.244	− 0.098	− 0.493
cosineQP	0.252	− 0.444	0.071	0.311	− −	− 0.755	− 0.672	− 0.536
sumIDFQP	− 0.193	− 0.104	− 0.223	0.396	− 0.245	− 0.068	− 0.474	− 0.725
avgIDFQP	− 0.209	− 0.029	− 0.222	0.369	− 0.305	0.056	− 0.557	− 0.652
maxIDFQP	− 0.105	− 0.084	− 0.029	0.37	− 0.308	− 0.232	− 0.048	− 0.372
sumICTFQP	0.02	− 0.01	− 0.052	0.418	− 0.162	− 0.011	− 0.263	− 0.619
avgICTFQP	0.007	0.121	− 0.042	0.534	− 0.288	0.204	− 0.413	− 0.542
maxICTFQP	− 0.263	0.059	− 0.535	0.367	− 0.36	− 0.253	− 0.176	− 0.372
SCSQP	− 0.002	0.185	− 0.028	0.494	− 0.323	0.322	− 0.43	− 0.446
sumSCQQP	− 0.134	− 0.269	− 0.135	0.144	− 0.032	− 0.261	− 0.34	− 0.755
avgSCQQP	− 0.302	− 0.11	− 0.169	0.223	− 0.243	0.1	− 0.681	− 0.565
maxSCQQP	− 0.001	− 0.052	–	–	–	–	–	–
sumVARQP	0.005	− 0.189	− 0.116	− 0.232	0.217	− 0.492	0.361	− 0.051
avgVARQP	− 0.019	− 0.17	− 0.137	− 0.313	0.237	− 0.509	0.48	0.124
maxVARQP	0.034	− 0.181	− 0.115	− 0.241	0.272	− 0.471	0.376	− 0.009
jointQP	0.005	− 0.189	− 0.116	− 0.232	0.217	− 0.492	0.361	− 0.051
joint2QP	0.034	− 0.181	− 0.115	− 0.241	0.272	− 0.471	0.376	− 0.009
profIDF	0	− 0.069	0.217	− 0.451	0.162	− 0.378	0.39	0.459
profICTF	− 0.145	− 0.256	− 0.011	− 0.57	0.192	− 0.537	0.29	0.301
profSCQ	0.051	0.041	0.144	− 0.359	0.094	− 0.366	0.451	0.424
profVAR	0.175	0.008	0.158	− 0.35	0.368	0.01	0.344	0.482

For '–' values *diffPerso* and predictor values are the same for all the evaluation triplets and given profile, therefore the standard deviation is zero and there is no correlation value

Table 10 *diffPerso*-predictors correlation values according to user profile and predictor for ASPIRE

	Administration	Agriculture	Culture	Economy	Education	Employment	Environment	Health
numQT	0.132	− 0.001	0.208	0.048	0.011	− 0.315	− 0.231	0.021
avgQTL	− 0.112	− 0.047	− 0.111	− 0.185	− 0.203	0.178	− 0.106	− 0.157
sumIDF	0.164	− 0.013	0.129	0.062	0.038	− 0.096	− 0.159	− 0.144
avgIDF	0.003	− 0.072	− 0.074	− 0.016	− 0.014	0.184	− 0.069	− 0.162
maxIDF	0.108	0.04	0.105	0.08	0.084	0.054	− 0.058	0.017
sumICTF	0.181	0.012	0.185	0.072	0.043	− 0.229	− 0.189	− 0.089
avgICTF	0.045	− 0.001	− 0.02	0.046	0.022	0.158	− 0.002	− 0.186
maxICTF	0.141	0.08	0.109	0.111	0.138	0.056	− 0.034	− 0.009
SCS	− 0.069	− 0.043	− 0.139	− 0.022	− 0.04	0.292	0.073	− 0.192
sumSCQ	0.154	− 0.022	0.186	0.047	− 0.003	− 0.304	− 0.251	− 0.054
avgSCQ	− 0.041	− 0.165	− 0.181	− 0.099	− 0.105	0.087	− 0.127	− 0.206
maxSCQ	0.069	0.005	0.028	0.039	− 0.088	− 0.02	− 0.142	0.011
sumVAR	0.02	− 0.057	0.137	− 0.026	0.043	− 0.017	0.038	0.057
avgVAR	0.047	− 0.051	0.141	− 0.014	0.084	0.066	0.102	0.125
maxVAR	0.018	− 0.057	0.137	− 0.025	0.05	− 0.016	0.057	0.072
joint	0.02	− 0.057	0.137	− 0.026	0.043	− 0.017	0.038	0.057
joint2	0.018	− 0.057	0.137	− 0.025	0.05	− 0.016	0.056	0.072
cosineQP	− 0.28	− 0.299	− 0.259	− 0.346	− 0.376	− 0.329	− 0.203	− 0.141
sumIDFQP	0.164	− 0.013	0.129	0.062	0.038	− 0.096	− 0.159	− 0.144
avgIDFQP	0.11	− 0.016	− 0.044	0.045	0.034	0.158	0.023	− 0.173
maxIDFQP	0.102	0.042	0.121	0.074	0.122	0.089	− 0.007	0.029
sumICTFQP	0.181	0.012	0.185	0.072	0.043	− 0.229	− 0.189	− 0.089
avgICTFQP	0.157	0.037	0.008	0.087	0.066	0.115	0.058	− 0.189
maxICTFQP	0.138	0.064	0.101	0.078	0.146	0.158	− 0.009	− 0.024
SCSQP	0.051	0.026	− 0.108	0.026	0.043	0.288	0.17	− 0.165
sumSCQQP	0.154	− 0.022	0.186	0.047	− 0.003	− 0.304	− 0.251	− 0.054
avgSCQQP	0.051	− 0.094	− 0.169	− 0.016	− 0.065	0.139	0.009	− 0.201
maxSCQQP	0.056	0.048	0.039	0.059	− 0.025	0.01	− 0.006	0.052
sumVARQP	0.025	− 0.056	0.187	− 0.023	0.04	− 0.109	− 0.018	0.08
avgVARQP	0.029	− 0.054	0.185	− 0.023	0.057	− 0.064	0.018	0.095
maxVARQP	0.017	− 0.058	0.137	− 0.027	0.047	− 0.039	0.055	0.077
jointQP	0.025	− 0.056	0.187	− 0.023	0.04	− 0.109	− 0.018	0.08
joint2QP	0.017	− 0.058	0.137	− 0.027	0.047	− 0.039	0.055	0.077
profIDF	0.007	0.076	0.076	0.025	0.019	− 0.185	0.078	0.159
profICTF	− 0.032	0.006	0.023	− 0.039	− 0.017	− 0.161	0.009	0.184
profSCQ	0.049	0.17	0.179	0.108	0.108	− 0.08	0.141	0.203
profVAR	− 0.05	0.048	− 0.132	0.011	− 0.087	− 0.086	− 0.118	− 0.128

References

Amati, G., Carpineto, C., Romano, G. (2004). Query difficulty, robustness, and selective application of query expansion. In *European conference on information retrieval* (pp. 127–137): Springer.

Aslam, J.A., & Pavlu, V. (2007). Query hardness estimation using jensen-shannon divergence among multiple scoring functions. In *European conference on information retrieval* (pp. 198–209): Springer.

Carmel, D., & Yom-Tov, E. (2010). Estimating the query difficulty for information retrieval. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, *2*(1), 1–89.

Carmel, D., Yom-Tov, E., Darlow, A., Pelleg, D. (2006). What makes a query difficult? In *Proceedings of the 29th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 390–397): ACM.

Chen, C., Yang, M., Li, S., Zhao, T., Qi, H. (2010). Predicting query potential for personalization, classification or regression? In *Proceedings of the 33rd international ACM SIGIR conference on research and development in information retrieval* (pp. 725–726): ACM.

Cronen-Townsend, S., Zhou, Y., Croft, W.B. (2002). Predicting query performance. In *Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 299–306): ACM.

Dan, O., & Davison, B.D. (2016). Measuring and predicting search engine users' satisfaction. *ACM Computing Surveys*, *49*(1), 18.

de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Romero, A.E. (2006). 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* (pp. 1024–1031).

de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Martín-dancausa, C., Romero, A.E. (2009). New utility models for the garnata information retrieval system at inex'08. In *Advances in focused retrieval* (pp. 39–45): Springer.

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

Dou, Z., Song, R., Wen, J.R. (2007). A large-scale evaluation and analysis of personalized search strategies. In *Proceedings of the 16th international conference on world wide web* (pp. 581–590): ACM.

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

Guo, Q., White, R.W., Dumais, S.T., Wang, J., Anderson, B. (2010). Predicting query performance using query, result, and user interaction features. In *Adaptivity, personalization and fusion of heterogeneous information* (pp. 198–201): Le Centre de hautes Etudes Internationales d'Informatique Documentaire.

Hauff, C. (2010). Predicting the effectiveness of queries and retrieval systems. Thesis, Centre for Telematics and Information Technology University of Twente.

Hauff, C., Hiemstra, D., de Jong, F. (2008a). A survey of pre-retrieval query performance predictors. In *Proceedings of the 17th ACM conference on information and knowledge management* (pp. 1419–1420): ACM.

Hauff, C., Murdock, V., Baeza-Yates, R. (2008b). Improved query difficulty prediction for the web. In *Proceedings of the 17th ACM conference on information and knowledge management* (pp. 439–448): ACM.

Hauff, C., Kelly, D., Azzopardi, L. (2010). A comparison of user and system query performance predictions. In *Proceedings of the 19th ACM international conference on information and knowledge management* (pp. 979–988): ACM.

He, B., & Ounis, I. (2004). Inferring query performance using pre-retrieval predictors. In *International symposium on string processing and information retrieval* (pp. 43–54): Springer.

He, J., Larson, M., De Rijke, M. (2008). Using coherence-based measures to predict query difficulty. In *European conference on information retrieval* (pp. 689–694): Springer.

Järvelin, K., & Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, *20*(4), 422–446.

Liu, F., Yu, C., Meng, W. (2004). Personalized web search for improving retrieval effectiveness. *IEEE transactions on Knowledge and Data Engineering*, *16*(1), 28–40.

Micarelli, A., Gasparetti, F., Sciarrone, F., Gauch, S. (2007). Personalized search on the world wide web. In *The adaptive web* (pp. 195–230): Springer.

Mothe, J., & Tanguy, L. (2005). Linguistic features to predict query difficulty. In *ACM conference on research and development in information retrieval, SIGIR, predicting query difficulty-methods and applications workshop* (pp. 7–10).

Shtok, A., Kurland, O., Carmel, D., Raiber, F., Markovits, G. (2012). Predicting query performance by query-drift estimation. *ACM Transactions on Information Systems*, *30*(2), 11.

Steichen, B., Ashman, H., Wade, V. (2012). A comparative survey of personalised information retrieval and adaptive hypermedia techniques. *Information Processing & Management*, *48*(4), 698–724.

Teevan, J., Dumais, S.T., Liebling, D.J. (2008). To personalize or not to personalize: modeling queries with variation in user intent. In *Proceedings of the 31st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 163–170): ACM.

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

Vicente-López, E., de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Tagua-Jiménez, A., Tur-Vigil, C. (2015). An automatic methodology to evaluate personalized information retrieval systems. *User Modeling and User-Adapted Interaction*, *25*(1), 1–37.

Vicente-López, E., de Campos, L.M., Fernández-Luna, J.M., Huete, J.F. (2016). Use of textual and conceptual profiles for personalized retrieval of political documents. *Knowledge-Based Systems*, *112*, 127–141.

Younus, A., Qureshi, M.A., O'Riordan, C., Pasi, G. (2013). Personalization for difficult queries. *SIGIR Workshop on Modeling User Behavior for Information Retrieval Evaluation*, 15–16.

Zhang, W., Wang, J., Chen, B., Zhao, X. (2013). To personalize or not: a risk management perspective. In *Proceedings of the 7th ACM conference on recommender systems* (pp. 229–236): ACM.

Zhao, Y., Scholer, F., Tsegay, Y. (2008). Effective pre-retrieval query performance prediction using similarity and variability evidence. In *European conference on information retrieval* (pp. 52–64): Springer,.

Zhou, Y., & Croft, W.B. (2006). Ranking robustness: a novel framework to predict query performance. In *Proceedings of the 15th ACM international conference on information and knowledge management* (pp. 567–574): ACM.

Zhou, Y., & Croft, W.B. (2007). Query performance prediction in web search environments. In *Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 543–550): ACM.

Zhou, B., & Yao, Y. (2010). Evaluating information retrieval system performance based on user preference. *Journal of Intelligent Information Systems*, *34*(3), 227–248.