Chapter 5

Performance prediction in Information Retrieval

Information retrieval performance prediction has been mostly addressed as a query performance issue, which refers to the performance of an information retrieval system in response to a specific query. It also relates to the appropriateness of a query as an expression of the user's information needs. In general, performance prediction methods have been classified into two categories depending on the used data: pre-retrieval approaches, which make the prediction before the retrieval stage using query features, and post-retrieval approaches, which use the rankings produced by a retrieval engine. In particular, the so-called clarity score predictor – of special interest for this thesis – has been defined in terms of language models, and captures the ambiguity of a query with respect to the utilised document collection, or a specific result set.

In this chapter we provide an overview of terminology, techniques, and evaluation related to performance prediction in Information Retrieval. In Section 5.1 we introduce terminology and foundamental concepts of the performance prediction problem. In Section 5.2 we describe the different types of performance prediction approaches, which are mainly classified in the two categories mentioned above: pre-retrieval and post-retrieval approaches. Then, in Section 5.3 we provide a thorough analysis on the use of clarity score as a performance prediction technique, including examples, adaptations, and applications found in the literature. Finally, in Section 5.4 we introduce the general methodology used to evaluate performance predictors, along with the most common methods to measure their quality.

5.1 Introduction

Performance prediction has received little attention, if any, to date in the Recommender Systems field. Our research, however, finds a close and highly relevant reference in the adjacent Information Retrieval discipline, where performance prediction has gained increasing attention since the late 90's, and has become an established research topic in the field. Performance prediction finds additional motivation in personalised recommendation, inasmuch the applications they are integrated in may decide to produce recommendations or hold them back, delivering only the sufficiently reliable ones. Moreover, the ability to predict the effectiveness of individual algorithms can be envisioned as a strategy to optimise the combination of algorithms into ensemble recommenders, which currently dominate the field – rarely if ever are individual algorithms used alone in working applications, neither are they found individually in the top ranks of evaluation campaigns and competitions (Bennett and Lanning, 2007).

In Information Retrieval performance prediction has been mostly addressed as a query performance problem (Cronen-Townsend et al., 2002). Query performance refers to the performance of an information retrieval system in response to a particular query. It also relates to the appropriateness of a query as an expression of a user's information needs. Dealing effectively with poorly-performing queries is a crucial issue in Information Retrieval since it could improve the retrieval effectiveness significantly (Carmel and Yom-Tov, 2010).

In general, performance prediction techniques can be useful from different perspectives (Zhou and Croft, 2006; Yom-Tov et al., 2005a):

- From the user's perspective, it provides valuable feedback that can be used to direct a search, e.g. by rephrasing the query or suggesting alternative terms.
- From the system's perspective, it provides a means to address the problem of information retrieval consistency. The consistency of retrieval systems can be addressed by distinguishing poorly performing queries. A retrieval system may invoke different retrieval strategies depending on the query, e.g. by using query expansion or ranking functions based on the predicted difficulty of the query.
- From the system administrator's perspective, it may let identify queries related to a specific subject that are difficult for the search engine. According to such queries, the collection of documents could be extended to better answer insufficiently covered topics.
- From a distributed information retrieval's perspective, it can be used to decide which search engine (and/or database) to use, or how much weight give to different search engines when their results are combined.

Specifically, the performance prediction task in Information Retrieval is formalised based on the following three core concepts: **performance predictor**, **retrieval quality assessment**, and **predictor quality assessment**. In this context, the performance predictor is defined as a function that receives the query (and the result list D_q retrieved by the system, the set of relevant documents R_q, collection statistics $\mathcal{C}$, etc.), and returns a prediction of the retrieval quality for that query. Then, by means of a predictor quality assessment method, the predictive power of the performance predictor is estimated.

Based on the notation given in (Carmel and Yom-Tov, 2010), the problem of performance prediction consists of estimating a true retrieval quality metric $\mu(q)$ (retrieval quality assessment) of an information retrieval system for a given query q. Hence, a performance predictor $\hat{\mu}(q)$ has the following general form:

$$\hat{\mu}(q) \leftarrow \gamma(q, R_q, D_q, \mathcal{C}) \tag{5.1}$$

The prediction methods proposed in the literature establish different functions γ, and use a variety of available data, such as the query's terms, its properties with respect to the retrieval space (Cronen-Townsend et al., 2002), the output of the retrieval system – i.e., D_q and R_q – (Carmel et al., 2006), and the output of other systems (Aslam and Pavlu, 2007).

According to whether or not the retrieval results are used in the prediction process, such methods can be classified into pre-retrieval and post-retrieval approaches, which are described in Sections 5.2.1 and 5.2.2, respectively. Another relevant distinction is based on whether the predictors are trained or not, but this classification is less popular, and will not be considered here.

Moreover, the standard methodology to measure the effectiveness of performance prediction techniques (that is, the predictor quality assessment method) consists of comparing the rankings of several queries based on their actual precision – in terms of a an evaluation metric such as MAP – with the rankings of those queries based on their performance scores, i.e., their predicted precision. In Section 5.4 we detail this methodology, along with several techniques for comparing the above rankings.

5.1.1 Notion of performance in Information Retrieval

In order to identify good performance predictors, validating or assessing their potential, we first have to define metrics of actual performance. Performance metrics and evaluation have been a core research and standardisation area for decades in the Information Retrieval field. In this section we introduce and summarise the main performance metrics and evaluation methodologies developed in the field.

The notion of performance in general, and in Information Retrieval in particular, leads itself to different interpretations, views and definitions. A number of methods for measuring performance have been proposed and adopted (Hauff et al., 2008a; Hauff, 2010), the most prominent of which will be summarised herein; see (Baeza-Yates and Ribeiro-Neto, 2011) for an extended discussion.

As a result of several decades of research by the Information Retrieval community, a set of standard performance metrics has been established as a consensual reference for evaluating the goodness of information retrieval systems. These metrics generally require a collection of documents and a query (or alternative forms of user input such as item ratings), and assume a ground truth notion of relevance – traditional notions consider this relevance as binary, while others, more recently proposed, consider different relevance degrees.

One of the simplest and widespread performance metrics in Information Retrieval is **precision**, which is defined as the ratio of retrieved documents that are relevant for a particular query. In principle, this definition takes all the retrieved documents into account, but can also consider a given cut-off rank as the **precision at n** or **P@n**, where just the top-n ranked documents are considered. Other related and widespread metric is **recall**, which is the fraction of relevant documents retrieved by the system. These two metrics are inversely related, since increasing one generally reduces the other. For this reason, usually, they are combined into a single metric – e.g. the **F-measure**, and the **Mean Average Precision** or **MAP** –, or the values of one metric are compared at a fixed value of the other metric – e.g. the **precision-recall curve**, which is a common representation that consists of plotting a curve of precision versus recall, usually based on 11 standard recall levels (from 0.0 to 1.0 at increments of 0.1).

An inherent problem of using MAP for poorly performing queries, and in general of any query-averaged metric, is that changes in the scores of better-performing queries mask changes in the scores of poorly performing queries (Voorhees, 2005b). For instance, the MAP of a baseline system in which the effectiveness is 0.02 for a query A, and 0.40 for a query B, is the same as the MAP of a system where query A doubles its effectiveness (0.04) and query B decreases a 5% (0.38). In this context, in (Voorhees, 2005a) two metrics were proposed to measure how well information retrieval systems avoid very poor results for individual queries: the **%no measure**, which is the percentage of queries that retrieved no relevant documents in the top 10 ranked results, and the **area measure**, which is the area under the curve produced by plotting MAP(X) versus X, where X ranges over the worst quarter queries. These metrics were shown to be unstable when evaluated in small sets of 50 queries (Voorhees, 2005b). A third metric was introduced in (Voorhees, 2006): **gmap**, the geometric mean of the average precision scores of the test set of queries. This metric emphasises poorly performing queries while it minimises differences between larger scores, remaining stable in small sets of queries (e.g. 50 queries) (Voorhees, 2005b).

Nonetheless, despite the above metrics and other efforts made to obtain better measures of query performance, MAP, and more specifically the **Average Precision per query**, are still widely used and accepted. See (Carmel et al., 2006; Cronen-Townsend et al., 2002; Hauff et al., 2008b; He and Ounis, 2004; He et al., 2008; Kompaoré et al., 2007; Zhao et al., 2008; Zhou and Croft, 2006; Zhou and Croft, 2007), among others.

Almost as important as the performance metric is the query type, which can be related to the differerent user information needs (Broder, 2002). Most work on performance prediction has focused on the traditional ad-hoc retrieval task where query performance is measured according to topical relevance (also known as content-based queries). Some work – such as (Plachouras et al., 2003) and (Zhou and Croft, 2007) – has also addressed other types of queries such as named page finding queries, i.e., queries focused on finding the most relevant web page assuming the queries contain some form of the "name" of the page being sought (Voorhees, 2002a).

When documents are timed (e.g. a newswire system), we can also distinguish two main types of queries that have been only partially exploited in the literature (Diaz and Jones, 2004; Jones and Diaz, 2007): those queries that favour very recent documents, and those queries for which there are more relevant documents within a specific period in the past.

Finally, we note that most of the research ascribed to predict performance has been focused not on predicting the "true" performance of a query (whatever that means), but on discriminating those queries where query expansion or relevance feedback algorithms have proved to be efficient from those where these algorithms fail, such as polisemic, ambiguous, and long queries. These are typically called *bad-to-expand* queries (Cronen-Townsend et al., 2006), illustrating the implicit dependence on their final application.

5.1.2 A taxonomy of performance prediction methods

Existing prediction approaches are typically categorised into pre-retrieval methods and post-retrieval methods (Carmel and Yom-Tov, 2010). Pre-retrieval methods make the prediction before the retrieval stage, and thus only exploit the query's terms and statistics about these terms gathered at indexing time. In contrast, post-retrieval methods use the rankings produced by a search engine, and, more specifically, the score returned for each document along with statistics about such documents and their vocabulary.

Category	Sub-category	Performance predictor (name and reference)
Pre-retrieval	Linguistics	Morphological, syntactic, semantic: (Mothe and Tanguy, 2005), (Kompaoré et al., 2007)
	Statistics	Coherency: coherence (He et al., 2008); term variance (Zhao et al., 2008) Similarity: collection query similarity (Zhao et al., 2008) Specificity: IDF-based (Plachouras et al., 2004), (He and Ounis, 2004); query scope (He and Ounis, 2004), (Macdonald et al., 2005); simplified clarity: (He and Ounis, 2004) Term relatedness: mutual information (Hauff et al., 2008a)
Post-retrieval	Clarity	Clarity (Cronen-Townsend et al., 2002), (Cronen-Townsend et al., 2006) Improved clarity (Hauff, 2010) (Hauff et al., 2008b) Jensen-Shannon Divergence (Carmel et al., 2006) Query difficulty (Amati et al., 2004)
	Robustness	Cohesion: clustering tendency (Vinay et al., 2006); spatial autocorrelation (Diaz, 2007); similarity (Kwok et al., 2004), (Grivolla et al., 2005) Document perturbation: ranking robustness (Zhou and Croft, 2006); document perturbation (Vinay et al., 2006) Query perturbation: query feedback (Zhou and Croft, 2007); autocorrelation (Diaz and Jones, 2004) (Jones and Diaz, 2007); query perturbation (Vinay et al., 2006); sub-query overlap (Yom-Tov et al., 2005a) Retrieval perturbation: (Aslam and Pavlu, 2007)
	Score analysis	Normalised Query Commitment: (Shtok et al., 2009) Standard deviation of scores: (Pérez-Iglesias and Araujo, 2009), (Cummins et al., 2011) Utility Estimation Framework: (Shtok et al., 2010) Weighted Information Gain: (Zhou and Croft, 2007)

Table 5.1. Overview of predictors presented in Section 5.2 categorised according to the taxonomy presented in (Carmel and Yom-Tov, 2010).

Pre-retrieval performance predictors do not rely on the retrieved document set, but on other information mainly extracted from the query issued by the user, such as statistics computed at indexing time (e.g. inverse term document frequencies). They have the advantage that predictions can be produced before the system's response is even started to be elaborated, which means that predictions can be taken into account to improve the retrieval process itself. However, they have a potential handicap with regards to their accuracy on the predictions, since extra retrieval effectiveness cues available with the system's response are not exploited (Zhou, 2007). Pre-retrieval query performance has been studied from two main perspectives: based on probabilistic methods (and more generally, on collection statistics), and based on linguistic approaches. Most research on the topic has followed the former approach. Some researchers have also explored inverse document frequency (IDF) and related features as predictors, along with other collection statistics

Post-retrieval performance predictors, on the other hand, make use of the retrieved results. Broadly speaking, techniques in this category provide better prediction accuracy compared to pre-retrieval performance predictors. However, many of these techniques suffer from high computational costs. Besides, they cannot be used to improve the retrieval strategies without a post-processing step, as the output from the latter is needed to compute the predictions in the first place. In (Carmel and Yom-Tov, 2010) post-retrieval methods are classified as follows: 1) clarity based methods that measure the coherence (clarity) of the result set and its separability from the whole collection of documents; 2) robustness based methods that estimate the robustness of the result set under different types of perturbations; and 3) score analysis based methods that analyse the score distribution of results.

Table 5.1 shows a number of representative approaches on performance prediction, which will be described in the next section. These approaches are categorised according to the taxonomy and sub-categories proposed in (Carmel and Yom-Tov, 2010). In the table we can observe that the statistics category has been the most popular approach for pre-retrieval performance prediction. Several predictors have been categorised in the robustness category, probably due to its broad meaning (query, document, and retrieval perturbation). Finally, we note that recent effort from the community has been focused on the score analysis category.

5.2 Query performance predictors

In this section we explain the distinct performance predictors proposed in the literature. As mentioned before, based on whether or not retrieval results are needed to compute performance scores, predictors can be classified into two main types: pre-retrieval and post-retrieval predictors. In the following we summarise some of the approaches of each of the above types. For additional information, the reader is referred to (Carmel and Yom-Tov, 2010), (Hauff, 2010), and (Pérez Iglesias, 2012).

5.2.1 Pre-retrieval predictors

Pre-retrieval performance predictors do not rely on the retrieved document set, and exploit other collection statistics, such as the inverse document frequency (IDF). In this context, performance prediction has been studied from three main perspectives: based on linguistic methods, based on statistical methods, and based on probabilistic methods.

Linguistic methods

In (Mothe and Tanguy, 2005) and (Kompaoré et al., 2007) the authors consider 16 query features, and study their correlation with respect to average precision and recall. These features are classified into three different types according to the linguistic aspects they model:

- Morphological features:
    - **Number of words**.
    - **Average word length** in the query.
    - **Average number of morphemes per word**, obtained using the CELEX[7] morphological database. The limit of this method is the database coverage, which leaves rare, new, and misspelled words as mono-morphemic.
    - **Average number of suffixed tokens**, obtained using the most frequent suffixes from the CELEX database (testing if each lemma in a topic is eligible for a suffix from this list).
    - **Average number of proper nouns**, obtained by POS (part-of-speech) tagger's analysis.
    - **Average number of acronyms**, detected by pattern matching.
    - **Average number of numeral values**, also detected by pattern matching.
    - **Average number of unknown tokens**, marked by a POS tagger. Most unknown words happen to be constructed words such as "mainstreaming", "postmenopausal" and "multilingualism."
- Syntactic features:
    - **Average number of conjunctions**, detected through POS tagging.
    - **Average number of prepositions**, also detected through POS tagging.
    - **Average number of personal pronouns**, again detected through POS tagging.
    - **Average syntactic depth**, computed from the results of a syntactic analyser. It is a straightforward measure of syntactic complexity in terms of

[7] CELEX, English database (1993). Available at www.mpi.nl/world/celex

hierarchical depth; it simply corresponds to the maximum number of nested syntactic constituents in the query.

- **Average syntactic links span**, computed from the results of a syntactic analyser; it is the average pairwise distance (in terms of number of words) between individual syntactic links.

- Semantic features:
    - **Average polysemy value**, computed as the number of synsets in the WordNet[8] database that a word belongs to, and averaged over all terms of the query.

In the above papers the authors investigated the correlation between these features, and precision and recall over datasets with different properties, and found that the only feature that positively correlated with the two performance metrics was the number of proper nouns. Besides, many variables did not obtain significant correlations with respect to any performance metric.

Statistical methods

Inverse document frequency is one of the most useful and widely used magnitudes in Information Retrieval. It is usually included in the information retrieval models to properly compensate how common terms are. Its formulation usually takes an ad hoc, heuristic form, even though formal definitions exist (Roelleke and Wang, 2008; Aizawa, 2003; Hiemstra, 1998). The main motivation for the inclusion of an IDF factor in a retrieval function is that terms that appear in many documents are not very useful for distinguishing a relevant document from a non-relevant one. In other words, it can be used as a measure of the specificity of terms (Jones, 1972), and thus as an indicator of their discriminatory power. In this way, IDF is commonly used as a factor in the weighting functions for terms in text documents. The general formula of IDF for a term t is the following:

$$\mathrm{IDF}(t) = \log \frac{N}{N_t} \tag{5.2}$$

where N is the total number of documents in the system, and n_t is the number of documents in which the term t appears.

Some research work on performance prediction has studied IDF as a basis for defining predictors. He and Ounis (2004) propose a predictor based on the **standard deviation of the IDF** of the query terms. Plachouras et al. (2004) represent the quality of a query term by a modification of IDF where instead of the number of documents, the number of words in the whole collection is used (**inverse collection term**

[8] WordNet, lexical database for the English language. Available at http://wordnet.princeton.edu/

frequency, or ICTF), and the query length acts as a normalising factor. These IDF-based predictors displayed moderate correlation with query performance.

Other authors have taken the similarity of the query into account. Zhao et al. (2008) compute the vector-space based query similarity with respect to the collection, considered as a large document composed of concatenation of all the documents. Then, different **collection query similarity** predictors are defined based on the SCQ values (defined below) for each query term, by summing, averaging, or taking the maximum values:

$$\mathrm{SCQ}(t) = (1 + \log \mathrm{TF}(t))\cdot \mathrm{IDF}(t) \tag{5.3}$$

The similarity of the documents returned by the query has also been explored in the field. The inter-similarity of documents containing query terms is proposed in (He et al., 2008) as a measure of **coherence**, by using the cosine similarity between every pair of documents containing each term. Additionally, two predictors based on the **pointwise mutual information** (PMI) are proposed in (Hauff et al., 2008a). The PMI of two terms is computed as follows:

$$\mathrm{PMI}(t_1, t_2) = \log \frac{p(t_1, t_2)}{p(t_1)p(t_2)} \tag{5.4}$$

where these probabilities can be approximated by maximum likelihood estimations, that is, based on collection statistics, where $p(t_1, t_2)$ is proportional to the number of documents containing both terms, and $p(t) \propto \mathrm{TF}(t)$. In that paper a first predictor is defined by computing the average PMI of every pair of terms in the query, whereas a second predictor is defined based on the maximum value. The predictive power of these techniques remains competitive, and is very efficient at run time.

Probabilistic methods

These methods measure characteristics of the retrieval inputs to estimate performance. He and Ounis (2004) propose a **simplified** version of the **clarity score** (see next section) in which the query model is estimated by the term frequency in the query:

$$\mathrm{SCS} = \sum_{w} P_{ml}(w|q) \log_2 \frac{P_{ml}(w|q)}{P(w|C)} \tag{5.5}$$

$$P_{ml}(w|q) = \frac{\mathrm{qtf}}{\mathrm{ql}}; P(w|C) = \frac{\mathrm{TF}(w)}{|V|}$$

where qtf is the number of occurrences of a query term w in the query, ql is the query length, $\mathrm{TF}(w)$ is the number of occurrences of a query term in the whole collection, and $|V|$ is the total number of terms in the collection.

Despite its original formulation, where the clarity score can be considered as a pre-retrieval predictor (Cronen-Townsend et al., 2002), Cronen-Townsend and colleagues use result sets to improve the computation time. For this reason, it is typically classified as a post-retrieval predictor (Zhou, 2007; Hauff et al., 2008a), and thus, we describe it with more detail in the next sections.

Kwok et al. (2004) build a query predictor using support vector regression, by training classifiers with features such as document frequencies and query term frequencies. In the conducted experiments they obtained a small correlation between predicted and actual query performances. He and Ounis (2004) propose the notion of **query scope** as a measure of the specificity of a query, which is quantified as the percentage of documents that contain at least one query term in the collection, i.e., $\log(N_Q/N)$, being N_Q the number of documents containing at least one of the query terms, and N the total number of documents in the collection. Query scope has shown to be effective in inferring query performance for short queries in ad hoc text retrieval, but very sensitive to the query length (Macdonald et al., 2005).

5.2.2 Post-retrieval predictors

Post-retrieval performance predictors make use of the retrieved results, in contrast to pre-retrieval predictions. Furthermore, computational efficiency is usually a problem for many of these techniques, which is balanced by better prediction accuracy. In the following we present the most representative approaches of each of the different sub-categories described in Section 5.1.2: clarity, robustness, and score analysis.

Clarity-based predictors

Cronen-Townsend et al. (2002) define **query clarity** as a degree of (the lack of) query ambiguity. Because of the particular importance and use of this predictor in the findings of this thesis, we shall devote a whole section (Section 5.3) for a thorough description and discussion about it. It is worth noting that the concept of query clarity has inspired a number of similar techniques. Amati et al. (2004) propose the **query difficulty** predictor to estimate query performance. In that work query performance is captured by the notion of the amount of information ($Info_{DFR}$) gained after the ranking. If there is a significant divergence in the query-term frequencies before and after the retrieval, then it is assumed that the divergence is caused by a query that is easy to respond to. $Info_{DFR}$ showed a significant correlation with average precision, but did not show any correlation between this predictor and the effectiveness of query expansion. The authors hence concluded that although the performance gains by query expansion in general increase as query difficulty decreases, very easy queries hurt the overall performance.

Adaptations of the query clarity predictor such as the one proposed in (Hauff et al., 2008b) will be discussed later in Section 5.3. Additionally, apart from the Kullback-Leibler divergence, the Jensen-Shannon Divergence on the retrieved document set and the collection also obtains a significant correlation between average precision and the distance measured (Carmel et al., 2006).

Robustness-based predictors

More recently, a related concept has been coined: **ranking robustness** (Zhou and Croft, 2006). It refers to a property of a ranked list of documents that indicates how stable a ranking is in the presence of *uncertainty* in its documents. The idea of predicting retrieval performance by measuring ranking robustness is inspired by a general observation in noisy data retrieval. The observation is that the degree of ranking robustness against noise is positively correlated with retrieval performance. This is because the authors assumed that regular documents also contain *noise*, if noise is interpreted as uncertainty. The robustness score performs better than, or at least as well as, the clarity score.

Regarding document and query perturbation, Vinay et al. (2006) propose four metrics to capture the geometry of the top retrieved documents for prediction: the **clustering tendency** as measured by the Cox-Lewis statistic, the sensitivity to **document perturbation**, the sensitivity to **query perturbation**, and the **local intrinsic dimensionality**. The most effective metric was the sensitivity to document perturbation, which is similar to the robustness score. Document perturbation, however, did not perform well for short queries, for which prediction accuracy dropped considerably when alternative state-of-the-art retrieval techniques (such as BM25 or a language modelling approach) were used instead of the TF-IDF weighting (Zhou, 2007).

Several predictors have been defined based on the concept of query perturbation. Zhou and Croft (2007) propose two performance predictors are defined based on this concept specifically oriented for Web search. First, the **Weighted Information Gain** predictor measures the amount of information gained about the quality of retrieved results (in response to a query) from an imaginary state that only an average document (represented by the whole collection) is retrieved to a posterior state that the actual search results are observed. This predictor was very efficient and showed better accuracy than clarity scores. The second predictor proposed in that work is the **Query Feedback**, which measures the degree of corruption that results from transforming Q to L (the output of the channel when the retrieval system is seen as a noisy channel, i.e., the ranked list of documents returned by the system). The authors designed a decoder that can accurately translate L back into a new query Q', whereupon the similarity between the original query Q and the new query Q' is taken as a performance predictor, since the authors interpreted the evaluation of the quality of the channel as the problem of predicting retrieval effectiveness. The computation of this predictor requires a higher computational cost than the previous one, being a major drawback of this technique.

Additionally, in (Diaz and Jones, 2004) and (Jones and Diaz, 2007) the authors exploited **temporal features** (time stamps) of the document retrieved by the query. They found that although temporal features are not highly correlated to performance, using them together with clarity scores improves prediction accuracy. Similarly, Diaz (2007) proposes to use the spatial autocorrelation as a metric to measure spatial similarities between documents in an embedded space, by computing the Moran's coefficient over the normalised scores of the documents. This predictor obtained good correlations results, although the author explicitly avoided collections such as question-answering and novelty related under the hypothesis that documents with high topical similarity should have correlated scores and, thus, in those collections the predictor would not work properly.

Other predictor was proposed in (Jensen et al., 2005), where visual features such as document titles and snippets are used from a surrogate document representation of retrieved documents. Such predictor was trained on a regression model with manually labelled queries to predict precision at the top 10 documents in Web search. The authors reported moderate correlation with respect to precision.

In (Yom-Tov et al., 2005a) two additional performance predictors are proposed. The first predictor builds a **histogram of the overlaps** between the results of each sub-query that agree with the full query. The second predictor is similar to the first one, but is based on a decision tree (Duda et al., 2001), which again uses overlaps between each sub-query and the full query. The authors apply these predictors to selective query expansion detecting missing content, and distributed information retrieval, where a search engine has to merge ranks obtained from different datasets. Empirical results showed that the quality of the prediction strongly depends on the query length.

The following predictors have been based on the cohesion of the retrieved documents. Kwok et al. (2004) propose predicting query performance by analysing similarities among retrieved documents. The main hypothesis of this approach is that relevant documents are similar to each other. Thus, if relevant documents are retrieved at the top ranking positions, the similarity between top documents should be high. The preliminary results, however, were inconclusive since negligible correlations were obtained. A similar approach is proposed in (Grivolla et al., 2005), where the entropy and pairwise similarity among top results are investigated. First, the entropy of the set of the K top-ranked documents for a query was computed. In this case it was assumed that the entropy should be higher when the performance for a given query is bad. Second, the mean cosine similarity between documents was proposed, using the base form of TF-IDF term weighting to define the document vectors. Correlation between average precision and the proposed predictors was not consistent along the different systems used in the experiment, although the predictors could still be useful for performance prediction, especially when used in combination.

Predictors based on score analysis

Finally, the last family of post-performance predictors analyses the score distributions of the results for each query. We have to note that the Weighted Information Gain predictor (Zhou and Croft, 2007) explained above is sometimes categorised into this group. In the following we present other predictors where the retrieved scores are explicit in the predictor computation.

For instance, the **Normalised Query Commitment** (NQC) predictor (Shtok et al., 2009) measures the standard deviation of the retrieval scores, and applies a normalisation factor based on the score of the whole collection:

$$\mathrm{NQC}(q) = \frac{\sqrt{1/|D_q| \sum_{d \in D_q} \left(s(d) - \mu_q\right)^2}}{|s(\mathcal{C})|} \tag{5.6}$$

where μ_q is the mean score of results in D_q (the retrieved set of documents for a query q). This predictor measures the divergence of results from their centroid, a "pseudo non-relevant document" that exhibits a relatively high query similarity (Carmel and Yom-Tov, 2010).

The **utility estimation framework** (UEF) was proposed in (Shtok et al., 2010) to estimate the utility of the retrieved ranking. In this framework three methods have to be specified to derive a predictor: a sampling technique for the document sets, a representativeness measure for relevance-model estimates, and a measure of similarity between ranked lists. Other authors have proposed approaches where standard deviation does not need to be computed for all the document scores in the retrieved results. Pérez-Iglesias and Araujo (2009) use a cutoff to decide how many documents are considered in the standard deviation computation. Moreover, Cummins et al. (2011) use different strategies to automatically select such cutoff.

Recently, Cummins (2012) has used Monte Carlo simulations to understand the correlations between average precision and the standard deviation of the scores in the head of a ranked list. The author found that the standard deviation of the list is positively correlated with the mean score of relevant documents, which in turn is positively correlated with average precision.

5.3 Clarity score

Cronen-Townsend et al. (2002) defined clarity score for Web retrieval as a measure of the lack of ambiguity of a particular query. More recently, it has been observed that this predictor also quantifies the diversity of the result list (Hummel et al., 2012). In this section we provide a deep analysis of this performance predictor since we shall use it along the rest of this thesis. We also describe examples and adaptations of the clarity score.

5.3.1 Definition of the clarity score

The clarity score predictor is defined as a Kullback-Leibler divergence between the query and the collection language model. It estimates the coherence of a collection with respect to a query q in the following way, given the vocabulary $\mathcal{V}$ and a subset of the document collection R_q consisting of those documents that contain at least one query term:

$$\mathrm{clarity}(q) = \sum_{w \in \mathcal{V}} p(w|q) \log_2 \frac{p(w|q)}{\mathrm{p}(w|\mathcal{C})} \tag{5.7}$$

$$p(d|q) = p(q|d)p(d)$$

$$p(q|d) = \prod_{w_q \in q} p(w_q|d)$$

$$p(w|q) = \sum_{d \in R_q} p(w|d)p(d|q)$$

$$p(w|d) = \lambda p_{\mathrm{ml}}(w|d) + \ (1-\lambda)p_{\mathrm{c}}(w)$$

The clarity value can thus be reduced to an estimation of the prior $\mathrm{p}(w|\mathcal{C})$ (collection language model), and the posterior $p(w|q)$ of the query terms w (query language model) using $p(w|d)$ over the documents $d \in R_q$ and based on term frequencies and smoothing. It should be emphasised that if the set R_q is chosen as the whole collection $\mathcal{C}$, then this technique could be classified as a pre-retrieval performance predictor, since no information about the retrieval would be used. The importance of the size of the relevance set R_q (or number of feedback documents) has been studied in (Hauff et al., 2008b), where an adaptation of the predictor was proposed in order to automatically set the number of documents to consider.

As first published in (Cronen-Townsend et al., 2002) and (Cronen-Townsend et al., 2006), query ambiguity is defined as "the degree to which a query retrieves documents in the given collection with similar word usage." Cronen-Townsend and colleagues found that queries whose highly ranked documents are a mix of documents from disparate topics receive lower scores than if they result in a topically-coherent retrieved set, and reported a strong correlation between the clarity score and the performance of a query. Because of that, the clarity score method has been widely used in the area for query performance prediction.

Some applications and adaptations of the clarity score metric include query expansion (anticipating poorly performing queries that should not be expanded), improving performance in the link detection task (more specifically, in topic detection and tracking by modifying the measure of similarity of two documents) (Lavrenko et al., 2002), and document segmentation (Brants et al., 2002). More applications can be found in Section 5.3.3.

Zhou (2007) provides a complementary formulation of the clarity score by rewriting the formulation used above as follows:

$$\text{clarity}(q) = \sum_{w \in V} \sum_{d \in R_q} p(w|d)p(d|q) \log \frac{\sum_{d \in R_q} p(w|d)p(d|q)}{p(w|\mathcal{C})} \tag{5.8}$$

In this way, Zhou emphasises, among other issues, the differences between the query clarity and the Weighted Information Gain predictor. Indeed, the author proposes the following generalisation of both formulations (for WIG and clarity). Specifically, the clarity formulation presented in Equations (5.7) and (5.8) is unified as follows:

$$\text{score}(q, \mathcal{C}, R) = \sum_{\xi \in T} \sum_{d \in R_q} \text{weight}(\xi, d) \log \frac{p(\xi, d)}{p(\xi, \mathcal{C})} \tag{5.9}$$

where T is a feature space, and R_q is a (ranked) document list. Besides this, $d \in R_q \subseteq \mathcal{C}$ must be comparable somehow with elements $\xi \in T$, in order to make sensible functions $\text{weight}(\xi, d)$ and $p(\xi, d)$. In this context, the query clarity as defined in (Cronen-Townsend et al., 2002) is an instantiation of Equation (5.9) where the following three aspects are considered:

- The feature space T is the whole vocabulary, consisting of single terms.
- The weight function is defined as $\text{weight}(\xi, d) = p(w|d)p(d|q)$.
- The function $p(\xi, d)$ is defined as $\sum_{d \in R_q} p(w|d)p(d|q)$, that is, it uses a document model averaged over all documents in the ranked list.

These observations help to discriminate between the underlying models used by these two predictors. In particular, for the query clarity, they also contribute to capture not so obvious divergences between a query and the collection, as we shall see in the next section.

train (0.33)	train dog (0.65)	obedience train dog (2.43)
		railroad train dog (0.67)
	railroad train (0.73)	railroad train caboose (1.46)

Table 5.2. Examples of clarity scores for related queries.

5.3.2 Interpreting clarity score in Information Retrieval

Aiming to better understand how the clarity score predictor behaves in Information Retrieval, and to what extent it is able to capture the difficulty or ambiguity of queries, in this section we summarise examples reported in the literature that let a clear interpretation of the predictor's values.

In a seminal paper (Cronen-Townsend et al., 2002) Cronen-Townsend and colleagues present the example shown in Table 5.2, which provides the clarity scores of a number of related queries that share some of their terms. These queries are related to each other in the sense that a particular query is formed by extending other query with an additional term, starting with an initial query formed by a single term, 'train' in the example. According to the queries of the table, we can observe that the term 'train' has different meanings for the largest queries; it refers to 'teach' in the query 'train dog', to the 'locomotive vehicle' in the query 'railroad train', and can refer to any of both meanings in the query 'railroad train dog.' The clarity scores capture the ambiguity of the queries (due to their different meanings for the term 'train'), independently from their length. In fact, the middle rightmost query 'railroad train dog' receives the lowest clarity score, corresponding to the most ambiguous query where the two considered meanings of 'train' are involved.

In the same paper, Cronen-Townsend and colleagues present the distribution of the language models for two queries, a clear query and a vague query (see Figure 2 in (Cronen-Townsend et al., 2002)). Each distribution is presented by plotting $p(w|q)\log_2 p(w|q)/\mathrm{p}(w|C)$ against the query terms w. The authors show that the distribution of the values of this function for the clear query dominates the distribution of the values of the vague query. This makes sense since the clarity score is computed by summing the probability values in the distribution of every term in the collection. Additionally, the authors show that the clear query presents spikes in its query language model when $p(w|q)$ is plotted against the terms, and compared with the collection probability $p(w|C)$. Hence, some of the terms with high contribution from the query language model (i.e., with high $p(w|q)$ values) obtain low collection probabilities ($p(w|C)$), thus evidencing a query that is different to the collection in its term usage (i.e., it is a non ambiguous query).

The above examples involve the (implicit) assumption known as *homogeneity assumption*, which specifies that the clarity score is higher if the documents in the considered collection are topically homogeneous. Hauff (2010) analyses the sensitivity of results with respect to that assumption. Specifically, the author computes the clarity score for three different ranked document lists: the relevant documents for a query, a non-relevant random sample, and a collection-wide random sample. The difference between the last two lists is that the second one is derived from documents judged as non-relevant, whereas the third one could contain any document in which at least one query term. Hauff shows how the clarity score is different depending on the origin of ranked document list, leading to a higher (lower) score by using relevant (non-relevant) documents for such list. However, we have to note that, as stated by Hauff, the quality in the separation of the clarity scores computed by each document list is different depending on the utilised dataset and queries.

The clarity score has been analysed in detail in Information Retrieval, mainly because its predictive power is superior to other performance predictors (in fact, it is one of the best performing post-retrieval predictors according to the overview presented in (Hauff, 2010)), but also because it provides interpretable results and high explanatory power in different IR processes, as we shall describe in the next section. Apart from that, the interest in this predictor is clear because of its probabilistic formulation and tight relationship with Language Models (Ponte and Croft, 1998).

5.3.3 Adaptations and applications of the clarity score

Cronen-Townsend and colleagues showed in (Cronen-Townsend et al., 2002) that clarity is correlated with performance, proving that the result quality is largely influenced by the amount of uncertainty involved in the inputs a system takes. In this sense, queries whose highly ranked documents belong to diverse topics receive lower scores than queries for which a topically-coherent result set is retrieved. Several authors have exploited the clarity score functionality and predictive capabilities (Buckley, 2004; Townsend et al., 2004; Dang et al., 2010), supporting its effectiveness in terms of performance prediction and high degree of adaptation. For instance, the predictor has been used for personalisation (Teevan et al., 2008) because of its proven capability of predicting ambiguity. In that paper the authors use more or less personalisation depending on the predicted ambiguity.

One of the first variants proposed in the area is the simplified clarity score proposed in (He and Ounis, 2004), presented in Section 5.2.1. In that paper He and Ouni changed the estimations of the posterior $p(w|q)$ to simple maximum likelihood estimators. Hauff et al. (2008b) proposed the Improved Clarity – called Adapted Clarity in (Hauff, 2010) –, in which the number of feedback documents

(R_q) is set automatically, and the term selection is made based on the frequency of the terms in the collection to minimise the contribution of terms with a high document frequency in the collection.

An alternative application of the clarity score is presented in (Allan and Raghavan, 2002), where the score obtained for the original set of documents returned by a query is compared against that obtained for a modified query, which was presumed to be more focused than the original one. Similarly, in (Buckley, 2004) Buckley uses the clarity score to measure the stability of the document rankings and compare it against a measure that uses the Mean Average Precision of each ranking (AnchorMap).

In (Sun and Bhowmick, 2009), Sun and Bhowmick adapted the concept of query clarity to image tagging, where a tag is visually representative if all the images annotated with that particular tag are visually similar to each other. In previous work (Sun and Datta, 2009) Sun and Datta proposed a similar concept, but in the context of blogging: a tag would receive a high clarity score if all blog posts annotated by the tag are topically cohesive.

Finally, an extension of the Kullback-Leibler divergence was proposed in (Aslam and Pavlu, 2007), where the Jensen-Shannon divergence was used instead. This distance is defined as the average of the Kullback-Leibler divergences of each distribution with respect to the average (or centroid) distribution. In this way, it is possible to compute the divergence between more than two distributions. Besides, the Jensen-Shannon divergence is symmetric, in contrast to the divergence used in the clarity score, and thus, a metric can be derived from it (Endres and Schindelin, 2003).

5.4 Evaluating performance predictors

In this section we describe the approaches proposed in the literature to evaluate the predictive power of a performance predictor. We define the different functions used to compute the quality of the performance predictors, most of them based on well known correlation coefficients between the true query performance values, and the expected or predicted performance values.

5.4.1 Task definition

Based on the notation presented in Section 5.1, in the following we present different techniques and functions to assess the effectiveness of performance predictors. Once the retrieval quality has been assessed ($\mu(q)$), and the value of the performance predictor for each query is calculated ($\hat{\mu}(q)$, using the function γ), the predictor quality is computed by using a predictor quality assessment function f^{qual} that measures the agreement between the true values of performance and the estimations, that is:

$$\text{Quality}(\gamma) = f^{qual}(\{\mu(q_1), \cdots, \mu(q_n)\}, \{\hat{\mu}(q_1), \cdots, \hat{\mu}(q_n)\}) \tag{5.10}$$

True quality values for each query are typically obtained by computing the per-query performance of a selected retrieval method (Cronen-Townsend et al., 2002; Hauff et al., 2008a), or by averaging the values obtained by several engines (Mothe and Tanguy, 2005), in order to avoid biases towards a particular method. As we shall see in the next section, the function f^{qual} typically represents a correlation coefficient; however, different possibilities are available and may be more appropriate depending on the prediction task.

In fact, in (Hauff et al., 2009) three estimation tasks were considered, by discriminating the output of the predictor function $\hat{\mu}$. **Query difficulty** estimation could be defined as a classification task where $\hat{\mu} \to \{0,1\}$ indicates whether the query is estimated to perform well or poorly. The standard estimation of **query performance**, nonetheless, would be defined by a function $\hat{\mu} \to \mathbb{R}$, in order to provide a ranking of queries, where the highest score denotes the best performing query. Furthermore, as stated in (Hauff et al., 2009), this function by itself does not directly estimate the performance metric μ. In order to do that we need to have normalised scores, such that the range of $\hat{\mu}$ is compatible with that of the metric, which typically requires $\hat{\mu} \to [0,1]$. In this case, we would be considering the **normalised query performance** task.

The methodology described above is general enough to be applicable to any of these three tasks, but is clearly inspired by the second one, that is, the estimation of query performance and it can be easily applied also to third one (normalised performance prediction). Because of that, we describe next a recently proposed methodology more focused on the (binary) query classification task or query difficulty prediction described in (Pérez-Iglesias and Araujo, 2010).

Let us suppose that, instead of continuous values of the performance metric μ, we are interested in estimating *as accurately as possible* the different difficulty grades of the queries, that is, $\mu \to \{1, \cdots, k\}$, where k is the number of difficulty grades available. Obviously, the output of the predictor $\hat{\mu}$ also has to be grouped in one of the k classes. Typically, we would have $k = 3$, representing "Easy", "Average", and "Hard" queries, although a binary partition could also be acceptable. In these terms the performance prediction problem is stated as a classification problem, where the goal is to effectively predict the query class.

Furthermore, this technique lets set, at the quality computation step, whether we want to weight uniformly each of the k classes, or if we are more interested in only one of them, by building, for instance, a confusion matrix, and applying standard Machine Learning evaluation metrics to a subset of it. In the next section we describe the most popular techniques for doing this, along with a new metric introduced in (Pérez-Iglesias and Araujo, 2010) oriented to the problem of performance prediction.

5.4.2 Measuring the quality of the predictors

There are several methods for measuring the quality of the performance prediction function $\hat{\mu}$ defined in the previous section. In particular, the quality function f^{qual} may be able to capture linear relations, take into account the importance implied by the scores or the ordering given by each variable (true and estimated performance, i.e., μ and $\hat{\mu}$), and exploit the implicit partitions derived by the method.

The most commonly used quality function is correlation, which has been measured by three well-known metrics: Pearson's, Spearman's, and Kendall's correlation coefficients. **Pearson's r correlation** captures linear dependencies between the variables, whereas **Spearman's ρ** and **Kendall's τ** correlation coefficients are used in order to uncover non-linear relationships between the variables. They are generally computed as follows, although in special situations (in presence of ties, or when there are missing values in the data) alternative formulations may be used:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{5.11}$$

$$\rho = 1 - \frac{6\sum_{i=1}^{n} d(x_i, y_i)^2}{n(n^2 - 1)} \tag{5.12}$$

$$\tau = 1 - \frac{4Q(x,y)}{n(n-1)} \tag{5.13}$$

where x and y represent the two variables of interest, $\bar{x}$ and $\bar{y}$ denote their means, $d(x_i, y_i)$ is the difference in ranks between x_i and y_i, and $Q(x,y)$ is the minimum number of swaps needed to convert the rank ordering of x to that of y. All these coefficients return values between -1 and $+1$, where -1 denotes a perfect anti-correlation, 0 denotes statistical independence, and $+1$ denotes perfect correlation.

It can be observed that Spearman's ρ computes a Pearson's r between the ranks induced by the scores of the variables. Moreover, Kendall's τ is the number of operations required to bring one list to the order of the other list using the *bubble sort* algorithm. Besides, although Spearman's and Kendall's correlations seem more general than Pearson's since they are able to capture non-parametric relations between the variables, we have to consider that distances between the scores are ignored in the rank-based coefficients, and thus, it is typically suggested to report one correlation coefficient of each type.

It is important to note that the number of points used to compute the correlation values affects the significance of the correlation results. The confidence test for a Pearson's r correlation, modeled as the t-value of a t-distribution (assuming normality) with $N - 2$ degrees of freedom (being N the size of the sample), is defined by the following equation (Snedecor and Cochran, 1989):

p-value	N		
	50	100	500
$p < 0.05$	1.677	1.661	1.648
$p < 0.01$	2.407	2.365	2.334

Pearson's r value	N		
	50	100	500
0.1	0.696	0.995	**2.243**
0.2	1.414	**2.021**	**4.555**
0.3	**2.179**	**3.113**	**7.018**
0.4	**3.024**	**4.320**	**9.739**

Table 5.3. Left: minimum t-value for obtaining a significant value with different sample sizes (N). Right: t-value for a given Pearson's correlation value and N points. In bold when the correlation is significative for $p < 0.05$, and underlined for $p < 0.01$.

$$t = r\sqrt{\frac{N-2}{1-r^2}} \tag{5.14}$$

The t-value therefore depends on the size of the sample, and thus, the significance of a Pearson's correlation value r may change depending on the number of test queries. In particular, for small samples, we may eventually obtain strong but non-significant correlations; whereas for large samples, on the other hand, we may obtain significant differences, even though the strength of the correlation values may be lower. The above also applies to the correlations computed using the Spearman's coefficient, but only under the null hypothesis or large sample sizes (greater than 100) (Snedecor and Cochran, 1989; Zar, 1972). For Kendall's correlation, the confidence test can be computed using an exact algorithm when there are no ties based on a power series expansion in N^{-1}, depending again, thus, on the sample size (Best and Gipps, 1974).

Table 5.3 shows the minimum t-value for obtaining a significant value with different sample sizes and p-values, along with the t-value computed using Equation (5.14) for different correlation values and sample sizes. In the table we can observe that the same correlation value may be significant or not depending on the size of the sample, for instance, with 50 queries, observations are significant with $p < 0.05$ for correlation values equal or above 0.3, whereas for 100 queries it is enough to obtain Pearson's correlation values of 0.2. This observation is related to the one presented in (Hauff et al., 2009), where Hauff and colleagues compared the confidence intervals of the three correlation coefficients described before, and observed how, due to the small query set sizes, most of the predictors analysed (pre-retrieval approaches such as clarity, IDF-based, and PMI) presented no significant differences, despite having very different values. In particular, this generated a subset of the analysed predictors that were not statistically different to the best performing predictor reported, and thus, any of the predictors in subset may be used in a later application since they obtain statistically similar (strictly speaking, not statistically different) correlations.

Furthermore, in the same paper, Hauff and colleagues proposed to use the **Root Mean Squared Error** (RMSE) as a quality function. The rationale behind this is that the RMSE squared is the function being minimised when performing a linear regression, and thus, it should also be able to capture the (linear) relation between the variables. In fact, there is a close relation between the RMSE and the Pearson's r coefficient, by means of the residual sum of squares (Carmel and Yom-Tov, 2010):

$$r^2 = 1 - \frac{SS_{err}}{SS_{tot}} = 1 - \frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{5.15}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n-1}} = \sqrt{\frac{SS_{err}}{n-1}} \tag{5.16}$$

Additional extensions to these correlation coefficients have been proposed. Most of these extensions have been focused on incorporating weights in the computation of the correlation (Melucci, 2009; Yilmaz et al., 2008). However, despite these metrics have an evident potential in the performance prediction area, to the best of our knowledge there is no work using them in order to evaluate the quality of the predictors (Pérez Iglesias, 2012).

Finally, a different family of quality functions can be considered in the query difficulty task, that is, when the performance prediction is cast as a classification problem. These techniques are based on the accuracy of the classification provided by the performance predictor, and thus, classic Machine Learning techniques could be used. In (Pérez-Iglesias and Araujo, 2010), Pérez-Iglesias and Araujo propose to use the **F-measure**:

$$F = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{5.17}$$

Additionally, in the same paper, Pérez-Iglesias and Araujo introduced a new metric (**distance based error measure**, or DBEM) along with a methodology that is focused on the misclassified difficulty classes between the predictor and the true classes. With this goal in mind, the authors apply a clustering algorithm to both the performance metric values and their estimations, aimed to minimise the distance between elements in the same group, and maximise the distance between elements in different groups. Specifically, Pérez-Iglesias and Araujo used the k-means algorithm, setting the value of k to the number of relevance grades, $k = 3$ in their paper. The metric DBEM is defined as follows:

$$\begin{gathered}\text{DBEM} = \frac{\sum_i^n \text{dist}\big(c(x_i), c(y_i)\big)}{\sum_i^n \max_j \text{dist}\Big(c(x_i), c\big(x_j\big)\Big)} \\ \text{dist}\big(c_i, c_j\big) = \|i - j\|, 0 < i, j \leq k\end{gathered} \tag{5.18}$$

where $c(x)$ is the function which assigns the proper class or partition to a given score x, according to the clustering algorithm. This metric captures the distance between every partition, normalised by the maximum possible distance. In this case, lower distances imply a better predictor quality.

5.5 Summary

Improvement of the predictive capabilities to infer the performance or difficulty of a query is consolidated as a major research topic in Information Retrieval, where it has been mostly applied to ad-hoc retrieval. Several performance predictors have been defined based on many different information sources, demonstrating the usefulness of such predictors in different tasks, mainly for query expansion, but also for rank fusion, distributed information retrieval, and text segmentation.

Some issues are, however, still open in the field, mostly regarding the evaluation of performance prediction. Performance prediction methods have been usually evaluated on traditional TREC document collections, which typically consist of no more than one million relatively homogenous newswire articles, and few research work has exploited these techniques with larger datasets; see, e.g. (Carmel et al., 2006; Zhou, 2007; Hauff, 2010) for some exceptions. Furthermore, reported correlation coefficient values have been typically computed using a small number of points (e.g. 50 queries for standard tracks in TREC), not always providing enough confidence to derive conclusions. And more importantly, how predictors have to be evaluated and which metric has to be used are still open research questions, that have generated some fruitful discussion in recent publications (Hauff, 2010; Pérez Iglesias, 2012), although a definitive answer has not been obtained yet.

We may presume that in the future other information retrieval applications may benefit from the framework derived by these techniques, and may develop tailored performance predictors by using purpose-designed performance metrics and evaluation methodologies, such as the recently developed concept of document difficulty in (Alvarez et al., 2012). This thesis is an example of such an application in the Recommender Systems field. More specifically, as we shall see in the next chapter, we translate the problem of performance prediction to the Recommender Systems area, where it has been barely studied. We focus our research on the query clarity predictor as a basis for the recommendation performance predictors, although additional techniques could be used, as we shall also present in Chapter 6. Finally, among the array of evaluation strategies presented above, we have decided to use correlations since it is the most common one in the literature, and provides a fair notion about the interpretability of the results.