Query Performance Prediction for IR

David Carmel, IBM Haifa Research Lab
Oren Kurland, Technion

SIGIR Tutorial
Portland Oregon, August 12, 2012

Instructors

Dr. David Carmel
- Research Staff Member at the Information Retrieval group at IBM Haifa Research Lab
- Ph.D. in Computer Science from the Technion, Israel, 1997
- Research Interests: search in the enterprise, query performance prediction, social search, and text mining

Dr. Oren Kurland
- Senior lecturer at the Technion — Israel Institute of Technology
- Ph.D. in Computer Science from Cornell University, 2006
- Research Interests: information retrieval
- kurland@ie.technion.ac.il, http://iew3.technion.ac.il/~kurland

Estimating the Query Difficulty — Main Challenge

- Even for systems that succeed very well on average, the quality of results returned for some of the queries is poor
- Understanding why some queries are inherently more difficult than others is essential for IR
- A good answer to this question will help search engines to reduce the variance in performance

Estimating the Query Difficulty — Some Benefits

- Feedback to users:
  - The IR system can provide the users with an estimate of the expected quality of the results retrieved for their queries.
  - Users can then rephrase “difficult” queries or, resubmit a “difficult” query to alternative search resources.
- Feedback to the search engine:
  - The IR system can invoke alternative retrieval strategies for different queries according to their estimated difficulty.
  - For example, intensive query analysis procedures may be invoked selectively for difficult queries only
- Feedback to the system administrator:
  - For example, administrators can identify missing content queries
  - Then expand the collection of documents to better answer these queries.
- For IR applications:
  - For example, a federated (distributed) search application
  - Merging the results of queries employed distributively over different datasets
  - Weighing the results returned from each dataset by the predicted difficulty

Contents

- Introduction — The Robustness Problem of (ad hoc) Information Retrieval
- Basic Concepts
- Query Performance Prediction Methods
  - Pre-Retrieval Prediction Methods
  - Post-Retrieval Prediction Methods
  - Combining Predictors
- A Unified Framework for Post-Retrieval Query-Performance Prediction
- A General Model for Query Difficulty
- A Probabilistic Framework for Query-Performance Prediction
- Applications of Query Difficulty Estimation
- Summary
- Open Challenges
Introduction – The Robustness Problem of Information Retrieval

The Robustness problem of IR

- Most IR systems suffer from a radical variance in retrieval performance when responding to users’ queries.
  - Even for systems that succeed very well on average, the quality of results returned for some of the queries is poor.
  - This may lead to user dissatisfaction.
- Variability in performance relates to various factors:
  - The query itself (e.g., term ambiguity "Golf")
  - The vocabulary mismatch problem - the discrepancy between the query vocabulary and the document vocabulary
  - Missing content queries - there is no relevant information in the corpus that can satisfy the information needs.

An example for a difficult query:
“The Hubble Telescope achievements”

The Reliable Information Access (RIA) workshop

- The first attempt to rigorously investigate the reasons for performance variability across queries and systems.
- Extensive failure analysis of the results of
  - 6 IR systems
  - 45 TREC topics
- Main reason to failures - the systems’ inability to identify all important aspects of the query
  - The failure to emphasize one aspect of a query over another, or emphasize one aspect and neglect other aspects
  - "What disasters have occurred in tunnels used for transportation?"
    - Emphasizing only one of these terms will deteriorate performance because each term on its own does not fully reflect the information need
- If systems could estimate what failure categories the query may belong to
  - Systems could apply specific automated techniques that correspond to the failure mode in order to improve performance

Table 1.1: RIA Topic Failure Analysis Categorization, Form 1(5)

Retrieved results deal with issues related to the Hubble telescope project in general, but the gist of that query, achievements, is lost.
Instability in Retrieval - The TREC's Robust Tracks

- The diversity in performance among topics and systems led to the TREC Robust tracks (2003 – 2005)
  - Encouraging systems to decrease variance in query performance by focusing on poorly performing topics
  - Systems were challenged with 50 old TREC topics found to be “difficult” for most systems over the years
  - A topic is considered difficult when the median of the average precision scores of all participants for that topic is below a given threshold
  - A new measure, GMAP, uses the geometric mean instead of the arithmetic mean when averaging precision values over topics
    Emphasizes the lowest performing topics, and is thus a useful measure that can attest to the robustness of system’s performance

How difficult is the performance prediction task for human experts?

- TREC 6 experiment: Estimating whether human experts can predict the query difficulty
- A group of experts were asked to classify a set of TREC topics to three degrees of difficulty based on the query expression only
  - easy, middle, hard?
- The manual judgments were compared to the median of the average precision scores, as determined after evaluating the performance of all participating systems
  - Results
    - The Pearson correlation between the expert judgments and the “true” values was very low (0.26).
    - The agreement between experts, as measured by the correlation between their judgments, was very low too (0.39)

The low correlation illustrates how difficult this task is and how little is known about what makes a query difficult

The Robust Tracks – Decreasing Variability across Topics

- Several approaches to improving the poor effectiveness for some topics were tested
  - Selective query processing strategy based on performance prediction
  - Post-retrieval reordering
  - Selective weighting functions
  - Selective query expansion
- None of these approaches was able to show consistent improvement over traditional non-selective approaches

- Apparently, expanding the query by appropriate terms extracted from an external collection (the Web) improves the effectiveness for many queries, including poorly performing queries

How difficult is the performance prediction task for human experts? (contd.)

Hauff et al. (CKM 2010) found
- a low level of agreement between humans with regard to which queries are more difficult than others (median kappa = 0.36)
- there was high variance in the ability of humans to estimate query difficulty although they shared “similar” backgrounds
- a low correlation between true performance and humans’ estimates of query-difficulty (performed in a pre-retrieval fashion)
  - median Kendall tau was 0.31 which was quite lower than that posted by the best performing pre-retrieval predictors
- however, overall the humans did manage to differentiate between “good” and “bad” queries
- a low correlation between humans’ predictions and those of query-performance predictors with some exceptions

These findings further demonstrate how difficult the query-performance prediction task is and how little is known about what makes a query difficult

The Robust Tracks – Query Performance Prediction

- As a second challenge: systems were asked to predict their performance for each of the test topics
- Then the TREC topics were ranked
  - First, by their predicted performance value
  - Second, by their actual performance value
- Evaluation was done by measuring the similarity between the predicted performance-based ranking and the actual performance-based ranking
- Most systems failed to exhibit reasonable prediction capability
  - 14 runs had a negative correlation between the predicted and actual topic rankings, demonstrating that measuring performance prediction is intrinsically difficult

On the positive side, the difficulty in developing reliable prediction methods raised the awareness of the IR community to this challenge

Are queries found to be difficult in one collection still considered difficult in another collection?

- Robust track 2005: Difficult topics in the ROBUST collection were tested against another collection (AQUAINT)
  - The median average precision over the ROBUST collection is 0.126
  - Compared to 0.185 for the same topics over the AQUAINT collection
  - Apparently, the AQUAINT collection is “easier” than the ROBUST collection probably due to
    - Collection use
    - Many more relevant documents per topic in AQUAINT
    - Document features such as structure and coherence
  - However, the relative difficulty of the topics is preserved over the two datasets
    - The Pearson correlation between topic’s performance in both datasets is 0.463
    - This illustrates some dependency between the topic’s median scores on both collections

Even when topics are somewhat easier in one collection than another, the relative difficulty among topics is preserved, at least to some extent
Basic concepts

The retrieval task
- Given:
  - A document set \( D \) (the corpus)
  - A query \( q \)
  - Retrieve \( D_q \) (the result list), a ranked list of documents from \( D \), which are most likely to be relevant to \( q \)

Some widely used retrieval methods:
- Vector space \( tf-idf \) based ranking, which estimates relevance by the similarity between the query and a document in the vector space
- The probabilistic OKAPI BM25 method, which estimates the probability that the document is relevant to the query
- Language-model-based approaches, which estimate the probability that the query was generated by a language model induced from the document
- And more, divergence from randomness (DFR) approaches, inference networks, markov random fields (MRF), ...

Text REtrieval Conference (TREC)
- A series of workshops for large-scale evaluation of (mostly) text retrieval technology:
  - Realistic test collections
  - Uniform, appropriate scoring procedures
  - Started in 1992
  
- A TREC task usually comprises of:
  - A document collection (corpus)
  - A list of topics (information needs)
  - A list of relevant documents for each topic (QRELS)

Prediction quality measures
- Given:
  - Query \( q \)
  - Result list \( D_q \) that contains \( n \) documents

Goal: Estimate the retrieval effectiveness of \( D_q \), in terms of satisfying \( I_q \), the information need behind \( q \)
  - Specifically, the prediction task is to predict \( AP(q) \) when no relevance information \( (R_q) \) is given.

In practice: Estimate, for example, the expected average precision for \( q \).
- The quality of a performance predictor can be measured by the correlation between the expected average precision and the corresponding actual precision values

Precision measures
- Precision at \( k \) (P@k): The fraction of relevant documents among the top-\( k \) results.
- Average precision: \( AP \) is the average of precision values computed at the ranks of each of the relevant documents in the ranked list:
  \[
  AP(q) = \frac{1}{|R_q|} \sum_{r \in R_q} P @ rank(r)
  \]
  \( R_q \) is the set of documents in the corpus that are relevant to \( q \)
  Average Precision is usually computed using a ranking which is truncated at some position (typically 1000 in TREC).

Measures of correlation
- Linear correlation (Pearson)
  - considers the true AP values as well as the predicted AP values of queries

- Non-linear correlation (Kendall’s tau, Spearman’s rho)
  - considers only the ranking of queries by their true AP values and by their predicted AP values

Title: African Civilian Deaths
Description: How many civilian non-combatants have been killed in the various civil wars in Africa?
Narrative: A relevant document will contain specific casualty information for a given area, country, or region. It will cite numbers of civilian deaths caused directly or indirectly by armed conflict.
Evaluating a prediction system (cond.)

- It is important to note that it was found that state-of-the-art query-performance predictors might not be correlated (at all) with measures of users' performance (e.g., the time it takes to reach the first relevant document).
  - see Terpin and Hersh ADC '04 and Zhao and Scholer ADC '07
- However, this finding might be attributed, as suggested by Terpin and Hersh in ADC '04, to the fact that standard evaluation measures (e.g., average precision) and users' performance are not always strongly correlated.
- Hersh et al. '00, Turpin and Hersh '01, Turpin and Scholer '06, Smucker and Parkash Jethani '10

Linguistic Approaches (Mothe & Tangui (SIGIR QD workshop, SIGIR 2005), Hauff (2010))

- Most linguistic features do not correlate well with the system performance.
  - Features include, for example, morphological (avg. # of morphemes per query term), syntactic link span (which relates to the average distance between query words in the parse tree), semantic (polysemy-the avg. # of synsets per word in the WordNet dictionary)
- Only the syntactic links span and the polysemy value were shown to have some (low) correlation.
- This is quite surprising as intuitively poor performance can be expected for ambiguous queries.
  - Apparently, term ambiguity should be measured using corpus-based approaches, since a term that might be ambiguous with respect to the general vocabulary, may have only a single interpretation in the corpus.
Pre-retrieval Statistical methods

- Analyze the distribution of the query term frequencies within the collection.
- Two major frequent term statistics:
  - Inverse document frequency \( idf(t) = \log(N/nt) \)
  - Inverse collection term frequency \( ictf(t) = \log(D/nt/D) \)
- Specificity based predictors
  - Measure the query terms’ distribution over the collection
  - Specificity-based predictors:
    - \( avgDF, avgCTF \): Queries composed of infrequent terms are easier to satisfy
    - \( maxDF, maxCTF \): Similarly
    - \( varDF, varCTF \): Low variance reflects the lack of dominant terms in the query
  - A query composed of non-specific terms is deemed to be more difficult
    - “Who and Whom”

Term Relatedness (Hauff 2010)

- Hypothesis: If the query terms co-occur frequently in the collection we expect good performance
- Pointwise mutual information (PMI) is a popular measure of co-occurrence statistics of two terms in the collection
  - It requires efficient tools for gathering collocation statistics from the corpus, to allow dynamic usage at query run-time
- \( \text{avgPMI}(t_1,t_2)/\text{maxPMI}(t_1,t_2) \) measures the average and the maximum PMI over all pairs of terms in the query

\[
PMI(t_1,t_2) \triangleq \log \frac{p(t_1,t_2 | D)}{p(t_1 | D)p(t_2 | D)}
\]

Evaluating pre-retrieval methods (Hauff 2010)

<table>
<thead>
<tr>
<th></th>
<th>ROBUST</th>
<th>T1ید</th>
<th>T2ید</th>
<th>T3ید</th>
<th>T4ید</th>
<th>GOV2</th>
<th>T1غี</th>
<th>T2غی</th>
<th>T3غی</th>
<th>T4غی</th>
</tr>
</thead>
<tbody>
<tr>
<td>( avgDF )</td>
<td>0.591</td>
<td>0.334</td>
<td>0.538</td>
<td>0.273</td>
<td>0.321</td>
<td>0.359</td>
<td>0.315</td>
<td>0.370</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( varDF )</td>
<td>0.178</td>
<td>0.119</td>
<td>0.131</td>
<td>0.087</td>
<td>0.189</td>
<td>0.132</td>
<td>0.278</td>
<td>0.096</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( maxDF )</td>
<td>0.122</td>
<td>0.167</td>
<td>0.124</td>
<td>0.429</td>
<td>0.393</td>
<td>0.473</td>
<td>0.371</td>
<td>0.366</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( avgICTF )</td>
<td>0.369</td>
<td>0.445</td>
<td>0.764</td>
<td>0.381</td>
<td>0.533</td>
<td>0.435</td>
<td>0.434</td>
<td>0.145</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( varICTF )</td>
<td>0.316</td>
<td>0.376</td>
<td>0.438</td>
<td>0.288</td>
<td>0.255</td>
<td>0.631</td>
<td>0.456</td>
<td>0.637</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Insights

- \( \text{maxVAR} \) and \( \text{maxSCQ} \) dominate other predictors, and are most stable over the collections and topic sets
- However, their performance significantly drops for one of the query sets (301-350)
- Prediction is harder over the Web collections (WT10G and GOV2) than over the news collection (ROBUST), probably due to higher heterogeneity of the data

Similarity, coherency, and variance-based predictors

- **Similarity**: SCQ (Zhao et al. 2008): High similarity to the corpus indicates effective retrieval \( SCQ(t) = ((1 + \log tf(t, D)) - idf(t)) \)
  - \( \text{maxSCQ} \) is the maximum/average over the query terms
- **Coherency**: CS (He et al. 2008): The average inter-document similarity between documents containing the query terms, averaged over query terms
  - This is a conceptual pre-retrieval reminiscent of the post-retrieval autocorrelation approach (Diaz 2007) that we will discuss later
  - Demanding computation that requires the construction of a pointwise similarity matrix for all pair of documents in the index
- **Variance** (Zhao et al. 2008): \( \text{var}(t) \) - Variance of term t’s weights (e.g., \( \text{tf.idf} \)) over the documents containing it
  - \( \text{maxVar} \) and \( \text{sumVar} \) are the maximum/average over query terms
  - Hypothesis: low variance implies to a difficult query due to low discriminative}
Post-Retrieval Predictors

- Analyze the search results in addition to the query
- Usually are more complex as the top results are retrieved and analyzed
- Prediction quality depends on the retrieval process
  - As different results are expected for the same query when using different retrieval methods
  - In contrast to pre-retrieval methods, the search results may depend on query-independent factors
  - Such as document authority scores, search personalization etc.
- Post-retrieval methods can be categorized into three main paradigms:
  - Clarity-based methods directly measure the coherence of the search results
  - Robustness-based methods evaluate how robust the results are to perturbations in the query, the result list, and the retrieval method
  - Score distribution based methods analyze the score distribution of the search results

Clarity (Cronen-Townsend et al. SIGIR 2002)

- Clarity measures the coherence (clarity) of the result-list with respect to the corpus
  - Good results are expected to be focused on the query’s topic
  - Clarity considers the discrepancy between the likelihood of words most frequently used in retrieved documents and their likelihood in the whole corpus
  - Good results - The language of the retrieved documents should be distinct from the general language of the whole corpus
  - Bad results - The language of retrieved documents tends to be more similar to the general language
- Accordingly, clarity measures the KL divergence between a language model induced from the result list and that induced from the corpus

Clarity Computation

\[
\text{Clarity}(q) = \sum_{t \in D_{t}} \frac{\log \frac{p(t | D_{q})}{p_{\text{MLE}}(t | X)}}{P_{\text{MLE}}(t | X)}
\]

- KL divergence between \( q \) and \( D \)
- \( P_{\text{MLE}}(t | X) \)’s unsmoothed LM (MLE)
- \( p(t | D_{q}) \)’s smoothed LM
- \( \lambda \)’s LM (RM1)

Query clarity score

Example

- Consider the two queries variants for TREC topic 56 (from the TREC query track):
  - Query A: Show me any predictions for changes in the prime lending rate and any changes made in the prime lending rates
  - Query B: What adjustments should be made once federal action occurs?

The Clarity of

1) relevant results, 2) non-relevant results, 3) random documents

Clarity Revisited - Hummel et al. 2012

(See the Clarity Revisited poster!)


\[
\text{Clarity}(q) = \text{Distance}(q) - \text{Diversity}(q)
\]

Cross entropy

Entropy
Robustness can be measured with respect to perturbations of the retrieval methods and measuring the diversity of the retrieved ranked lists.

Documents retrieval as a communication channel problem

- Small random perturbations of the document representation are unlikely to result in significant changes to the ranking

Retrieval method

- In general, different retrieval methods tend to retrieve different results for the same query, when applied over the same document collection
- A high overlap in results retrieved by different methods may be related to high agreement on the (usually sparse) set of relevant results for the query.
- A low overlap may indicate no agreement on the relevant results; hence, query difficulty

Robustness

- Robustness can be measured with respect to perturbations of the query
  - The robustness of the result list to small modifications of the query (e.g., perturbation of term weights)
  - Documents
    - Small random perturbations of the document representation are unlikely to result in major changes to documents' retrieval scores
    - If scores of documents are spread over a wide range, then these perturbations are unlikely to result in significant changes to the ranking

Cohesion of the Result list – Clustering tendency

- The cohesion of the result list can be measured by its clustering patterns
  - Following the "cluster hypothesis" which implies that documents relevant to a given query are likely to be similar to one another
  - A good retrieval returns a single, tight cluster, while poor retrieval returns a loosely related set of documents covering many topics
- The "clustering tendency" of the result set
  - Corresponds to the Cox-Lewis statistic which measures the "randomness" level of the result list
  - Measured by the distance between a randomly selected document and its nearest neighbor from the result list.
  - When the list contains "inherent" clusters, the distance between the random document and its closest neighbor is likely to be much larger than the distance between this neighbor and its own nearest neighbor in the list

Query Perturbations

- Overlap between the query and its sub-queries (Yom Tov et al. SIGIR 2005)
  - Observation: Some query terms have little or no influence on the retrieved documents, especially in difficult queries.
  - The query feedback (QF) method (Zhou and Croft SIGIR 2007) models retrieval as a communication channel problem
    - The input is the query, the channel is the search system, and the set of results is the noisy output of the channel
    - A new query is generated from the list of results, using the terms with maximal contribution to the Clarity score, and then a second list of results is retrieved for that query
    - The overlap between the two lists is used as a robustness score.

Retrieval Method Perturbation (Aslam & Pavlu, ECIR 2007)

- Query difficulty is predicted by submitting the query to different retrieval methods and measuring the diversity of the retrieved ranked lists
  - Each ranking is mapped to a distribution over the document collection
  - JSD distance is used to measure the diversity of these distributions
- Evaluation: Submissions of all participants to several TREC tracks were analyzed
  - The agreement between submissions highly correlates with the query difficulty, as measured by the median performance (AP) of all participants
  - The more submissions are analyzed, the prediction quality improves.
The linear correlation of the regularized scores with the original scores is used for query-performance prediction.

Spatial autocorrelation (Diaz SIGIR 2007)

- Query performance is correlated with the extent to which the result list “respects” the cluster hypothesis.
- The extent to which similar documents receive similar retrieval scores.
- In contrast, a difficult query might be detected when similar documents are scored differently.
- A document’s “regularized” retrieval score is determined based on the weighted sum of the scores of its most similar documents.

\[
\text{Score}_{\text{reg}}(q,d) = \sum_{d'} \text{Sim}(d,d') \cdot \text{Score}(q,d')
\]

The linear correlation of the regularized scores with the original scores is used for query-performance prediction.

Weighted Information Gain –WIG (Zhou & Croft 2007)

- WIG measures the divergence between the mean retrieval score of top-ranked documents and that of the entire corpus.
- The more similar these documents are to the query, with respect to the corpus, the more effective the retrieval.
- The corpus represents a general non-relevant document.

\[
\text{WIG}(q) = \sum_{d \in D} \sum_{d' \in D'} \lambda_d \log \left( \frac{p(t|d)}{p(t|D)} \right) \left( \text{avg}(\text{Score}(d')) - \text{Score}(D) \right)
\]

Evaluating post-retrieval methods (Shtok et al. TOIS 2012)

- QF and NQC exhibit comparable results.
- QF performs well over some of the collections but is inferior to other predictors on ROBUST.

QF performs well over some of the collections but is inferior to other predictors on ROBUST.
Additional predictors that analyze the retrieval scores distribution

- Computing the standard deviation of retrieval scores at a query-dependent cutoff
  - Perez-Iglesias and Araujo SPIRE 2010
  - Cummins et al. SIGIR 2011
- Computing expected ranks for documents
  - Vinay et al. CIKM 2008
- Inferring AP directly from the retrieval score distribution
  - Cummins AIRS 2011

Utility estimation framework (UEF) for query performance prediction (Shotk et al. SIGIR 2010)

- Suppose that we have a true model of relevance, $R_q$, for the information need $I_q$ that is represented by the query $q$
- Then, the ranking $\pi(D_q, R_q)$, induced over the given result list using $R_q$, is the most effective for these documents
- Accordingly, the utility (with respect to the information need) provided by the given ranking, which can be thought of as reflecting query performance, can be defined as

$$U(D_q | I_q) \equiv \text{Similarity}(D_q, \pi(D_q, R_q))$$

- In practice, we have no explicit knowledge of the underlying information need and of $R_q$
- Using statistical decision theory principles, we can approximate the utility by estimating $R_q$

$$U(D_q | I_q) \approx \int \text{Similarity}(D_q, \pi(D_q, R_q)) p(R_q | I_q) dR_q$$

Instantiating predictors

- Relevance-model estimates ($\hat{R}_q$)
  - Relevance language models constructed from documents sampled from the highest ranks of some initial ranking
- Estimate the relevance-model presumed “representativeness” of the information need ($\hat{R}_q$)
  - Apply previously proposed predictors (Clarity, WIG, QF, NQC) upon the sampled documents from which the relevance model is constructed
- Inter-list similarity measures
  - Pearson, Kendall's-tau, Spearman
- A specific, highly effective, instantiated predictor:
  - Construct a single relevance model from the given result list, $D_q$
  - Use a previously proposed predictor upon $D_q$ to estimate relevance-model effectiveness
  - Use Pearson's correlation between retrieval scores for the similarity measure

Combining Predictors

Combining post-retrieval predictors

- Some integration efforts of a few predictors based on linear regression:
  - Yom-Tov et al (SIGIR'05) combined avgIDF with the Overlap predictor
  - Zhou and Croft (SIGIR'07) integrated WIG and QF using a simple linear combination
  - Diaz (SIGIR'07) incorporated the spatial autocorrelation predictor with Clarity and with the document perturbation based predictor
- In all those trials, the results of the combined predictor were much better than the results of the single predictors
  - This suggests that these predictors measure (at least semi) complementary properties of the retrieved results
A unified post-retrieval prediction framework

We want to predict the effectiveness of a ranking \( \sigma_q(q, D) \) of the corpus that was induced by retrieval method \( M \) in response to query \( q \).

Assume a true model of relevance \( R_q \) that can be used for retrieval \( \Rightarrow \)
the resultant ranking \( \sigma_q(q, D) \), is the optimal utility

\[
\text{Utility}(\sigma_q(q, D), I) = \sum_{q \in D} \text{sim}(\sigma_q(q, D), x_q(q, D))
\]

Use as reference comparisons a pseudo effective (PE) and pseudo ineffective (PIE) rankings (i.e. ROCExp71):

\[
\text{Utility}(\sigma_q(q, D), I) = \alpha(q) \sum_{q \in D} \text{sim}(\sigma_q(q, D), x_q(q, D)) - \beta(q) \sum_{q \in D} \text{sim}^{-1}(\sigma_q(q, D), x_q(q, D))
\]

\[
\text{Ranking}(x_q(q, D)) \Rightarrow \text{Pseudo-effective ranking}(\alpha(q)\text{PE})
\]

\[
\text{Ranking}(x_q(q, D)) \Rightarrow \text{Pseudo-ineffective ranking}(\beta(q)\text{PIE})
\]

Instantiating predictors

Focus on the result lists of the documents most highly ranked by each ranking

\[
\hat{U}(\sigma_q(q, D), I) = \alpha(q) \sum_{q \in D} \text{sim}(\sigma_q(q, D), I_q) - \beta(q) \sum_{q \in D} \text{sim}^{-1}(\sigma_q(q, D), I_q)
\]

Deriving predictors:
1. "Guess" a PE result list \( (I_{PE}) \) and/or a PIE result list \( (I_{PIE}) \)
2. Select weights \( \alpha(q) \) and \( \beta(q) \)
3. Select an inter-list (ranking) similarity measure
   - Pearson's \( r \), Kendall's \( \tau \), Spearman's \( \rho \)

Using a Pseudo Ineffective (PIE) result list

\[
\hat{U}(\sigma_q(q, D), I) = \alpha(q) \sum_{q \in D} \text{sim}(\sigma_q(q, D), I_q) - \beta(q) \sum_{q \in D} \text{sim}^{-1}(\sigma_q(q, D), I_q)
\]

Basic idea: The PIE result list is composed of 6 copies of a pseudo ineffective document

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Pseudo Ineffective document</th>
<th>Predictor's description</th>
<th>Sim. measure</th>
<th>( \alpha(q) )</th>
<th>( \beta(q) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarity (Brown-Townsend 92-94)</td>
<td>Corpus</td>
<td>Estimates the focus of the result list with respect to the corpus</td>
<td>-1 to 1 down between language models</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>UEF (Weighted information Gain; Dash &amp; Croft 97)</td>
<td>Corpus</td>
<td>Measures the difference between the retrieval scores of documents in the result list and the score of the corpus</td>
<td>-1 to 1 distance of retrieval scores</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>NDC (Normalised Discounted Cumulative Gain; Croft et al. 1998)</td>
<td>Result list centroid</td>
<td>Measures the standard deviation of the retrieval scores of documents in the result list</td>
<td>-1 to 1 distance of retrieval scores</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
**SIGIR 2012 Tutorial: Query Performance Prediction**

### Using a Pseudo Effective (PIE) Result List

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Pseudo Effective result list</th>
<th>Predictors description</th>
<th>Sim. measure</th>
<th>α</th>
<th>β</th>
</tr>
</thead>
<tbody>
<tr>
<td>QE (Query Feedback; Zhou &amp; Croft '07)</td>
<td>Use a relevance model for retrieval over the corpus</td>
<td>Measures the “amount of noise” (non-query-related aspects) in the result list</td>
<td>Overlap at top ranks</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>LRF (Likelihood Estimation Framework; Shok et al. '08)</td>
<td>Revise the given result list using a relevance model</td>
<td>Estimates the potential utility of the result list using relevance models</td>
<td>Rank/Scores correlation</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Autocorrelation (Diaz '07)</td>
<td>1.Score regularization 2.Fusion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### A Theoretical Model of Topic Difficulty

**Main Hypothesis**
Topic difficulty is induced from the distances between the model parts.

**Model validation:** The Pearson correlation between Average Precision and the model distances (see the paper for estimates of the various distances)

Based on the .gov2 collection (25M docs) and 100 topics of the Terabyte tracks 04/05

### A General Model for Query Difficulty

**A model of query difficulty** (Carmel et al, SIGIR 2006)

- A user with a given information need (a topic):
  - Submits a query to a search engine
  - Judges the search results according to their relevance to this information need.
- Thus, the query/ies and the Qrels are two sides of the same information need.
- Qrels also depend on the existing collection (C)

**Define:** Topic = (Q,R|C)

### A probabilistic framework for QPP (Kurland et al. CIKM 2012, to appear)

- The basic question that underlies the QPP task:
  - What is the probability that this result list (D_q) is relevant to this query (q)?

\[
p(q,h_q,D_q) = \frac{p(D_q|h_q)p(h_q|C)}{p(D_q|C)}
\]

**Post-retrieval prediction**

- Normalizer
- Pre-retrieval prediction

---

**SIGIR 2012 Tutorial: Query Performance Prediction**
Post-retrieval prediction (Kurland et al. 2012)

\[
\pi(\mathbf{b}_{\pi} | \mathbf{q}, \mathbf{r}) = \sum_{\mathbf{d}_{\pi}} \pi(\mathbf{d}_{\pi} | \mathbf{r}) \pi(\mathbf{r} | \mathbf{q}, \mathbf{b}_{\pi})
\]

Assuming (some) uniform priors:

\[
\pi(\mathbf{d}_{\pi} | \mathbf{r}) = \pi(\mathbf{d}_{\pi}) \sum_{\mathbf{b}_{\pi}} \pi(\mathbf{b}_{\pi} | \mathbf{r}) \pi(\mathbf{r} | \mathbf{q}, \mathbf{b}_{\pi})
\]

query-independent result list properties (e.g., cohesion/dispersion)

WIG and Clarity are derived from this expression.

Main applications for QDE

- Feedback to the user and the system
- Federation and metasearch
- Content enhancement using missing content analysis
- Selective query expansion/query selection
- Others

Using reference lists (Kurland et al. 2012)

Using reference result lists:

\[
\pi(\mathbf{d}_{\pi} | \mathbf{r}) = \sum_{\mathbf{d}_{\ref}} \pi(\mathbf{d}_{\ref} | \mathbf{r}) \pi(\mathbf{r} | \mathbf{q}, \mathbf{d}_{\ref})
\]

Association (e.g., similarity) between the ideal list \(\mathbf{d}_{\ref}\) and the reference list \(\mathbf{d}_{\ref}\)

The presumed quality of the reference list (a prediction task at its own right)

- All predictors that use reference lists (except for UTIP) ignore \(\pi(\mathbf{d}_{\ref} | \mathbf{r})\)
- If a single reference list \(\mathbf{d}_{\ref}\) is used, and a symmetric inter-data similarity measure is employed for \(\pi(\mathbf{d}_{\ref} | \mathbf{r})\), as in the case, for example, for QF and ausoso relevance, then the prediction is not only for \(\mathbf{d}_{\ref}\) but also for \(\mathbf{d}_{\ref}\)

Feedback to the user and the system

- Direct feedback to the user
  - "We were unable to find relevant documents to your query"
- Estimating the value of terms from query refinement
  - Suggest which terms to add, and estimate their value
- Personalization
  - Which queries would benefit from personalization? (Teevan et al. ‘08)

Applications of Query Difficulty Estimation

Federation and metasearch: Problem definition

- Given several databases that might contain information relevant to a given question,
- How do we construct a good unified list of answers from all these datasets?
- Similarly, given a set of search engines employed for the same query,
- How do we merge their results in an optimal manner?
Prediction-based Federation & Metasearch

Weight each result set by its predicted precision (Yom-Tov et al., 2005; Sheldon et al., 11) or select the best predicted SE (White et al., 2008; Berger & Savoy, 2007).

Federation for the TREC 2004 Terabyte track

GOV2 collection: 426GB, 25 million documents

50 topics

For federation, divided into 10 partitions of roughly equal size

<table>
<thead>
<tr>
<th>Method</th>
<th>P@10</th>
<th>MAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>One collection</td>
<td>0.522</td>
<td>0.292</td>
</tr>
<tr>
<td>Prediction-based feder</td>
<td>0.550</td>
<td>0.264</td>
</tr>
<tr>
<td>Score-based federation</td>
<td>0.498</td>
<td>0.257</td>
</tr>
</tbody>
</table>

* 10-fold cross-validation

Metasearch and Federation using Query Prediction

- Train a predictor for each engine/collection pair
- For a given query: Predict its difficulty for each engine/collection pair
- Weight the results retrieved from each engine/collection pair accordingly, and generate the federated list

Content enhancement using missing content analysis

- A helpdesk system where system administrators try to find relevant solutions in a database
- The system administrators’ queries are logged and analyzed to find topics of interest that are not covered in the current repository
- Additional content for topics which are lacking is obtained from the internet, and is added to the local repository
Adding content where it is missing improves precision

- Cluster user queries
- Add content according to:
  - Lacking content
  - Size
  - Random
- Adding content where it is missing brings the best improvement in precision

Selective query expansion

- Automatic query expansion: Improving average quality by adding search terms
- Pseudo-relevance feedback (PRF): Considers the (few) top-ranked documents to be relevant, and uses them to expand the query.
- PRF works well on average, but can significantly degrade retrieval performance for some queries.
- PRF fails because of Query Drift: Non-relevant documents infiltrate into the top results or relevant results contain aspects irrelevant of the query's original intention.
- Rocchio's method: The expanded query is obtained by combining the original query and the centroid of the top results, adding new terms to the query and reweighting the original terms.

\[ Q_{e} = aQ_{original} + \beta \sum_{D_{i} \in top} D_{i} + \gamma \sum_{D_{i} \in unexpanded} D_{i} \]

Other uses of query difficulty estimation

- Collaborative filtering (Bellogin & Castells, 2009): Measure the lack of ambiguity in a users' preference.
- Term selection (Kumar & Carvalho, 2009): Identify irrelevant query terms.
- Queries with fewer irrelevant terms tend to have better results.
- Reducing long queries (Cummins et al., 2011)

Some work on selective query expansion

- Expand only "easy" queries
  - Predict which are the easy queries and only expand them (Amati et al., 2004; Yom-Tov et al., 2005)
  - Estimate query drift (Cronen-Townsend et al., 2006)
  - Compare the expanded list with the unexpanded list to estimate if too much noise was added by using expansion
- Selecting a specific query expansion form from a list of candidates (Winaver et al. 2007)
- Integrating various query-expansion forms by weighting them using query-performance predictors (Soskin et al., 2009)
- Estimate how much expansion (Lv and Zhai, 2009)
  - Predict the best \(\alpha\) in the Rochio formula using features of the query, the documents, and the relationship between them
- But, see Azzopardi and Hauff 2009

Summary & Conclusions

Summary

- In this tutorial, we surveyed the current state-of-the-art research on query difficulty estimation for IR
- We discussed the reasons that cause search engines to fail for some of the queries.
- Bring about a high variability in performance among queries as well as among systems.
- We summarized several approaches for query performance prediction.
- Pre-retrieval methods
- Post-retrieval methods
- Combining predictors
- We reviewed evaluation metrics for prediction quality and the results of various evaluation studies conducted over several TREC benchmarks.
- These results show that state-of-the-art existing predictors are able to identify difficult queries by demonstrating a reasonable prediction quality.
- However, prediction quality is still moderate and should be substantially improved in order to be widely used in IR tasks.
Summary – Main Results

- Current linguistic-based predictors do not exhibit meaningful correlation with query performance
  - This is quite surprising as intuitively poor performance can be expected for ambiguous queries
- In contrast, statistical pre-retrieval predictors such as SumSCQ, and maxVAR have relatively significant predictive ability
  - These are retrieval predictors, and a few others, exhibit comparable performance to post-retrieval methods such as Clarity, WIG, NGC, QF over large scale Web collections
  - This is counter-intuitive as post-retrieval methods are exposed to much more information than pre-retrieval methods
- However, current state-of-the-art predictors still suffer from low robustness in prediction quality
  - This robustness problem, as well as the moderate prediction quality of existing predictors, are two of the greatest challenges in query difficulty prediction, that should be further explored in the future

What’s Next?

- Predicting the performance for other query types
  - Navigational Queries
  - XML queries (Xquery, Xpath)
  - Domain Specific queries (e.g. Healthcare)
- Considering other factors that may affect query difficulty
  - Who is the person behind the query? in what context?
  - Geo-spatial features
  - Temporal aspects
  - Personal parameters
- Query difficulty in other search paradigms
  - Multifaceted search
  - Exploratory Search

Summary (cont)

- We tested whether combining several predictors together may improve prediction quality
  - Especially when the different predictors are independent and measure different aspects of the query and the search results
- An example for a combination method is using linear regression
  - The regression task is to learn how to optimally combine the predicted performance values in order to best fit them to the actual performance values
  - Results were moderate, probably due to the sparseness of the training data, which over-represents the lower end of the performance values
- We discussed three frameworks for query-performance prediction
  - Utility Estimation Framework (UEF), Shtok et al. 2010
  - A unified framework for post-retrieval prediction that sets common grounds for various previously-proposed predictors, Hurland et al. 2011
  - A fundamental framework for estimating query difficulty; Carmel et al. 2006

Concluding Remarks

- Research on query difficulty estimation has begun only ten years ago with the pioneering work on the Clarity predictor (2002)
- Since then this subfield has found its place at the center of IR research
  - These studies have revealed alternative prediction approaches, new evaluation methodologies, and novel applications
- In this tutorial we covered
  - Existing performance prediction methods
  - Some evaluation studies
  - Potential applications
  - Some anticipations on future directions in the field
- While the progress we see is enormous already, performance prediction is still challenging and far from being solved
  - Much more accurate predictors are required in order to be widely adopted by IR tasks
  - We hope that this tutorial will contribute in increasing the interest in query difficulty estimation

Summary (cont)

- We discussed a few applications that utilize query difficulty estimators
  - Handling each query individually based on its estimated difficulty
  - Find the best terms for query refinement by measuring the expected gain in performance for each candidate term
  - Expand the query or not based on predicted performance of the expanded query
  - Personalize the query selectively only in cases that personalization is expected bring value
  - Collection enhancement guided by the identification of missing content queries
  - Fusion of search results from several sources based on their predicted quality

Thank You!