The 7 Post-Retrieval Query Measures

Property	Measure	Description	Formula or Algorithm
Robustness	*Subquery Overlap*	Captures the extent of the overlap between the result set retrieved by the entire query and the result sets retrieved by individual query terms. The lower the standard deviation, the better the query.	i) Run the original query q, and obtain the result list R ii) Run each individual query term q_t in the original query as a separate query and obtain the result list R_t. iii) For each individual query term q_t, compute the overlap between the first k (k=10) documents in R and the first k documents in R_t (i.e., number of documents found in both result lists) iv) The overall score of the query is the standard deviation of the values of the overlap considered for each term in the query.
	Robustness Score	The terms' weights in the top relevant documents are slightly perturbed and the resulting documents are re-ranked. The correlation between the initial rank and that after modification is considered. The higher the robustness score, the better the query.	i) Run the original query q, and obtain the result list R ii) Take the top 50 documents in R and consider them as ranked list L iii) For each document d in L, get a perturbed document d' from d in the following way: • All terms t from the corpus that do not appear in document d, will not be included in d' neither. • All terms t from the corpus that appear in document d with frequency f, but do not appear in the query will appear in document d' with the same frequency f. • Each term t that appears in d with frequency f and appears also in the query q will appear also in d' , but with a frequency f', which is a random number obtained from a Poisson distribution P(λ) with λ = f iv) The new 50 documents obtained are ranked according to the query q, resulting in a second ranked list L', where each document corresponds to a document in L. v) Compute the Spearman rank correlation between the positions of the 50 documents in L and the positions of their corresponding perturbed documents in L' and record the correlation obtained. vi) Repeat steps iii) – v) 100 times, and the final robustness score is the average Spearman correlation between the 100 runs.
	First Rank Change	Captures the probability of a document found on the first position in the list of results to still remain on the first position after a perturbation is applied to it. The higher the score, the better the query.	i) Run the original query q, and obtain the result list R ii) Take the top 50 documents in R and consider them as ranked list L iii) For each document d in L, get a perturbed document d' from d in the following way: • All terms t from the corpus that do not appear in document d, will not be included in d' neither • All terms t from the corpus that appear in document d with frequency f, but do not appear in the query will appear in document d' with the same frequency f. • Each term t that appears in d with frequency f and appears also in the query q will appear also in d' , but with a frequency f', which is a random number obtained from a Poisson distribution P(λ) with λ = f iv) The new 50 documents obtained are ranked according to the query q, resulting in a second ranked list L', where each document corresponds to a document in L. v) record a 1 if the top ranked document in L is also the top ranked document (after perturbation) in L', and record 0 otherwise. vi) Repeat steps iii) – v) 100 times, and the final score is the sum of the values (0 or 1) obtained in step v) for all the 100 runs.
	Clustering Tendency	Measures the cohesion of the top retrieved documents as the textual similarity between them. The higher the clustering tendency the better the query.	$$CT = Mean\left(\frac{Sim_{query}(d_{mp}, d_{nn}\|q)}{Sim_{query}(p_{sp}, d_{mp}\|q)}\right) * \frac{1}{T}\sum_{i=1}^{T}(x_i - y_i)$$ q - the query p_{sp} – the sampled point, i.e., a randomly chosen document from the corpus, which does not appear in the top 100 documents d_{mp} - the marked point, i.e., the document, inside the top 100 documents in the ranked list , with largest similarity with the sampled point d_{nn} - the nearest neighbor of the marked point within the top 100 documents

			in the ranked list x_i - the maximum weight for a term i across the top 100 retrieved documents y_i - the minimum weight for a term i across the top 100 retrieved documents The mean in the CT formula is computed for 100 randomly sampled points (i.e., the similarity formulas are computed 100 times, each time with a different randomly sampled point). $$Sim_{query}(d_i, d_j \mid q) = \frac{\sum_{K=1}^{T} d_{ik} d_{jk}}{\sqrt{\sum_{k=1}^{T} d_{ik}^2 \sum_{k=1}^{T} d_{jk}^2}} * \frac{\sum_{k=1}^{T} c_k q_k}{\sqrt{\sum_{k=1}^{T} c_k^2 \sum_{k=1}^{T} q_k^2}}$$ d_i and d_j - the two documents T - the number of unique terms in the collection q - the query (with weight q_k for term k) - the weight of a term in the query or a document is its tfidf c - the vector of terms common to both d_i and d_j with weights c_k being the average of d_{ik} and d_{jk}.
	Spatial Autocorrelation	Changes the retrieval-scores of each top relevant document as the average of the scores of its most similar documents. Then, the linear correlation of the new scores with original ones is used. The higher the spatial autocorrelation the better the query.	i) Run the original query q, and obtain the result list R ii) Take the top 50 documents in R and consider them as ranked list L iii) For each document d in L, compute the cosine similarity between d and the rest of the documents in L, using tf-idf as the weight of the terms in the document vectors. iv) Among the documents in L, select the 5 documents that are most similar to d according to the cosine similarity. v) Let s be the score of document d in L. Assign a new score to d, which is the average score of the 5 most similar documents to it according to the cosine similarity. vi) Perform the above steps for each document d in L vii) The Pearson correlation between the original scores of the documents in L and the derived scores of those documents represents the index of spatial autocorrelation.
Score distribution	*Weighted Information Gain (WIG)*	Measures the divergence between the mean retrieval score of top-ranked documents and that of the entire corpus. The hypothesis is that the more similar these documents are to the query, with respect to the query similarity exhibited by a general non-relevant document (i.e., the corpus), the more effective the retrieval. The higher the weighted information gain, the better the query.	$$WIG(q) = \frac{1}{k} \sum_{d \in D_q^k} \sum_{t \in q} \lambda(t) log \frac{\Pr(t \mid d)}{\Pr(t \mid D)}$$ q – query t – a term in the query q D – set of all documents in corpus Dq – the set of documents in the result set to query q D_q^k – the top k documents in the result list k – the number of top documents to consider |q| - number of terms in the query q $\lambda(t) = \frac{1}{\sqrt{\|q\|}}$
	Normalized Query Commitment)NQC)	Measures the standard deviation of retrieval scores in D_q^k, normalized by the score of the whole collection. The higher NCQ, the better the query.	$$NCQ = \frac{\sqrt{\frac{1}{k} \sum_{d \epsilon D_q^k} (Score(d) - \mu)^2}}{Score(D_q)}$$ k – the number of top documents to consider. Best performance was obtained with k=100. D_q^k – The top k documents from the result list returned in response to query q. Score (d) – the score obtained by document d in D_q^k D_q – the set of all results returned in response to query q Score (D_q) – the sum of the scores of all the documents in the result list returned by the IR technique $\mu = \frac{1}{k} \sum_{d \epsilon D_q^k} Score(d)$ NCQ - represents the normalized standard deviation of the retrieval scores in D_q^k