# IR Evaluation

April 28, 2015

## 1 IR Ranking, Search Engine Output

### 1.1 comparing search engines

Web search engines have their ancestors in the information retrieval (IR) systems developed during the last fifty years. IR methods include (among others) the Boolean search methods, the vector space methods, the probabilistic methods, and the clustering methods [BelCroft87]. All these methods aim at finding the relevant documents for a given query.

One of the primary distinctions made in the evaluation of search engines is between effectiveness and efficiency.Effectiveness, loosely speaking, measures the ability of the search engine to find the right information, and efficiency measures how quickly this is done. For a given query, and a specific definition of relevance, we can more precisely define effectiveness as a measure of how well the ranking produced by the search engine corresponds to a ranking based on user relevance judgments. Efficiency is defined in terms of the time and space requirements for the algorithm that produces the ranking.Carrying out this type of holistic evaluation of effectiveness and efficiency, while important, is very difficult because of the many factors that must be controlled. For this reason, evaluation is more typically done in tightly defined experimental settings and this is the type of evaluation we focus on here.

To measure ad hoc information retrieval effectiveness in the standard way, we need a test collection consisting of three things:

1. A document collection
2. A test suite of information needs, expressible as queries
3. A set of relevance judgments, standardly a binary assessment of either relevant or non-relevant for each query-document pair.

Given these ingredients, how is system effectiveness measured? The two most frequent and basic measures for information retrieval effectiveness are precision(the number of relevant retrieved documents divided by the number of retrieved documents) and recall(the number of relevant retrieved documents divided by the number of relevant documents). One main use is in the TREC (Text retrieval conference, http://trec.nist.gov), where many research groups get their system tested against a common database of documents.

## 2 Set measures

There are several matrices that are used to measure the effectiveness of the IR system. The matrices are True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN).

First, Let us take a look at precision and recall in more detail. As an example, in an information retrieval scenario, the instances are documents and the task is to return a set of relevant documents given a search term; or equivalently, to assign each document to one of two categories, "relevant" and "not relevant". In this case, the "relevant" documents are simply those that belong to the "relevant" category. Recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents, while precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search. We have the formula for precision and recall as follows:

$$Precision = \frac{num(relevant\ items\ retrieved)}{num(retrieved\ items)} = P(relevant|retrieved) \tag{1}$$

$$Recall = \frac{num((relevant\ items\ retrieved))}{num((relevant\ items))} = P(retrieved|relevant) \tag{2}$$

These notions can be made clear by examining the *confusion matrix.*Given a ranking of documents, we can create a *confusion matrix* that counts the correct and incorrect answers of each type.

| | Relevant | Non-Relevant |
|---|---|---|
| Retrived | TP | FP |
| Non Retrived | FN | TN |

Table 1: Confusion Matrix

- True Positives(TP) are relevant documents in the ranking
- False Positives(FP) are non-relevant documents in the ranking
- True Negatives(TN) are non-relevant documents missing from the ranking
- False Negatives(FN) are relevant documents missing from the ranking

Now, we can express precison and recall in terms of confusion matrices terms:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

An obvious alternative that may occur to the reader is to judge an information retrieval system by its *accuracy*, that is, the fraction of its classification that are correct. In terms of the contingency table above,

$$accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \tag{5}$$

This seems plausible, since there are two actual classes, relevant and non-relevant, and an information retrieval system can be thought of as a two-class classifier which attempts to label them as such (it retrieves the subset of documents which it believes to be relevant). This is precisely the effectiveness measure often used for evaluating machine learning classification problems. There is a good reason why accuracy is not an appropriate measure for information retrieval problems. In almost all circumstances, the data is extremely skewed: normally over 99.9% of the documents are in the non-relevant category. A system tuned to maximize accuracy can appear to perform well by simply deeming all documents non-relevant to all queries. Even if the system is quite good, trying to label some documents as relevant will almost always lead to a high rate of false positives. However, labeling all documents as non-relevant is completely unsatisfying to an information retrieval system user. Users are always going to want to see some documents, and can be assumed to have a certain tolerance for seeing some false positives providing that they get some useful information. The measures of precision and recall concentrate the evaluation on the return of true positives, asking what percentage of the relevant documents have been found and how many false positives have also been returned.

A single measure that trades off precision versus recall is the *F measure*,which is the weighted harmonic mean of precision and recall:

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R} \quad where \;\; \beta^2 = \frac{1-\alpha}{\alpha} \tag{6}$$

where $\alpha \in [0,1]$ and thus $\beta^2 \in [0,\infty]$.The default *balanced F measure* equally weights precision and recall, which means making $\alpha = \frac{1}{2}$or $\beta = 1$. It is commonly written as $F_1$, which is a short of $F_\beta = 1$. When using $\beta = 1$, the formula on the right simplifies to :

$$F_{\beta=1} = \frac{2PR}{P+R} \tag{7}$$

Values of $\beta < 1$ emphasize precision, while values of $\beta > 1$ emphasize recall. For example, a value of $\beta = 3$ or $\beta = 5$ might be used if recall is to be emphasized. Recall,precision, and the F measure are inherently measures between 0 and 1, but they are also very commonly written as percentages, on a scale between 0 and 100.

# 3 Ranking Measures

Precision, recall, and the F measure are set-based measures. They are computed using unordered sets of documents. We need to extend these measures (or to define new measures) if we are to evaluate the ranked retrieval results that are now standard with search engines. In a ranked retrieval context, appropriate sets of retrieved documents are naturally given by the top k retrieved documents. For each such set, precision and recall values can be plotted to give a precision-recall curve as shown in figure 1.

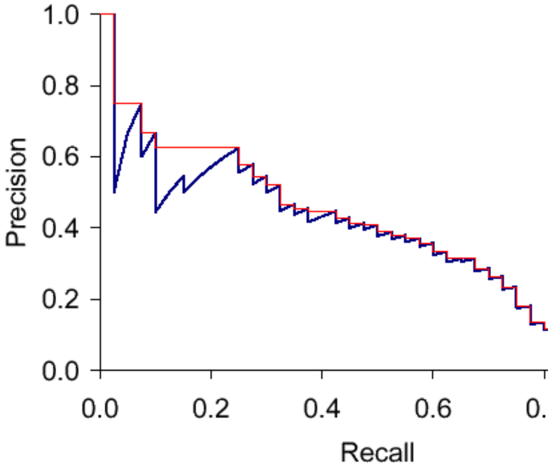

Figure 1: Precision-Recall Curve

The above measures factor in precision at all recall levels. For many prominent applications, particularly web search, this may not be germane to users. What matters is rather how many good results there are on the first page or the first three pages. This leads to measuring precision at fixed low levels of retrieved results, such as 10 or 30 documents. This is referred to as *"Precision at k"*, for example "Precision at 10". It has the advantage of not requiring any estimate of the size of the set of relevant documents but the disadvantage that it is the least stable of the commonly used evaluation measures and that it does not average well, since the total number of relevant documents for a query has a strong influence on precision at k.

Another way to model user behavior is based on the probability that document i is the last document read. This gives an interpretation for *Average Precision*: the expected relevance gained from the user choosing a relevant document i uniformly at random, and reading all documents from 1 to i. Imagine that exactly one of the relevant documents will satisfy the user, but we don't know which one.

$$L_M(i) := \frac{P_M(i) - P_M(i+1)}{P_M(1)} \tag{8}$$

An alternative, which alleviates this problem, is *R-precision.* It requires having a set of known relevant documents Rel, from which we calculate the precision of the top Rel documents returned. (The set Rel may be incomplete, such as when Rel is formed by creating relevance judgments for the pooled top k results of particular systems in a set of experiments.) R-precision adjusts for the size of the set of relevant documents: A perfect system could score 1 on this metric for each query, whereas, even a perfect system could only achieve a precision at 20 of 0.4 if there were only 8 documents in the collection relevant to an information need. If there are |Rel| relevant documents for a query, we examine the top |Rel| results of a system, and find that r are relevant, then by definition, not only is the precision (and hence R-precision) r/|Rel|, but the recall of this result set is also r/|Rel|.

The *Reciprocal Rank* of a query response is the multiplicative inverse of the rank of the first correct answer. The mean reciprocal rank (MRR) is the average of the reciprocal ranks of results for a sample of queries Q:

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{9}$$

For example, suppose we have the following three sample queries for a system that tries to translate English words to their plurals. In each case, the system makes three guesses, with the first one being the one it thinks is most likely correct:

| Query | Results | Correct response | Rank | Reciprocal Rank |
|---|---|---|---|---|
| cat | catten,cati,**cats** | cats | 3 | 1/3 |
| torus | torii,**tori**,toruses | tori | 2 | 1/2 |
| virus | **viruses**,virii,viri | viruses | 1 | 1 |

Given those three samples, we could calculate the mean reciprocal rank as $(1/3 + 1/2 + 1)/3 = 11/18$ or about 0.61.

## 3.1 Receiver Operating Characteristics curve (ROC)

Another concept sometimes used in evaluation is an ROC curve. An ROC curve plots the true positive rate or sensitivity against the false positive rate or (1 - specificity). Here, sensitivity is just another term for recall. The false positive rate is given by $fp/(fp + tn)$. Figure 2 shows the ROC curve corresponding to the precision-recall curve in Figure 1. An ROC curve always goes from bottom left to the top right of the graph. For a good system, the graph climbs steeply on the left side. Precison-recall curves are sometimes loosely refered to as ROC curve. This is understand, but not accurate.

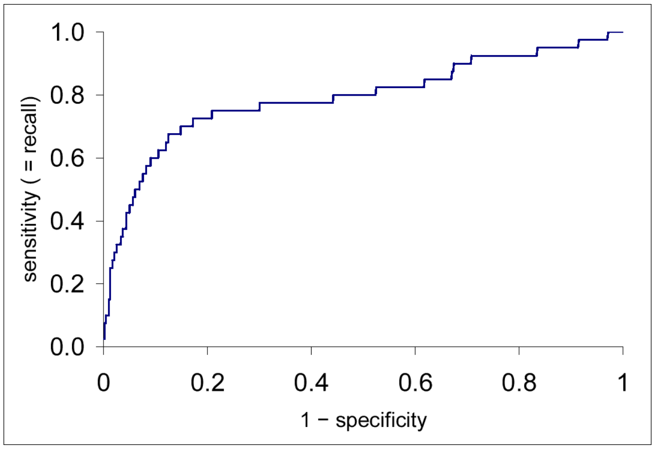

Figure 2: ROC Curve

## 3.2 nDCG and DCG

A final approach that has seen increasing adoption, especially when employed with machine learning approaches to ranking is measure of cumulative gain and in particular *normalized discounted cumulative gain* (NDCG) . NDCG is designed for situations of non-binary notions of relevance . Like precision at k, it is evaluated over some number k of top search results. For a set of queries Q, let R(j,d) be the relevance scores assessors gave to document d for query j. Then,

$$NDCG(Q,k) = \frac{1}{|Q|}\sum_{j=1}^{|Q|} Z_{kj} \sum_{m=1}^{k} \frac{2^{R(j,m)}-1}{log_2(1+m)} \tag{10}$$

where $Z_{kj}$ is a normalization factor calculated to make it so that a perfect rankings NDCG at k for query j is 1. For queries for which $k' < k$ documents are retrieved, the last summation is done up to k'.

Discounted cumulative gain (DCG) is a measure of ranking quality. In information retrieval, it is often used to measure effectiveness of web search engine algorithms or related applications. Using a graded relevance scale of documents in a search engine result set, DCG measures the usefulness, or gain, of a document based on its position in the result list. The gain is accumulated from the top of the result list to the bottom with the gain of each result discounted at lower ranks.

Two assumptions are made in using DCG and its related measures.

1. Highly relevant documents are more useful when appearing earlier in a search engine result list (have higher ranks)
2. Highly relevant documents are more useful than marginally relevant documents, which are in turn more useful than irrelevant documents.

The premise of DCG is that highly relevant documents appearing lower in a search result list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result. The discounted CG accumulated at a particular rank position p is defined as:

$$DCG_p = rel_1 + \sum_{i=2}^{|p|} \frac{rel_i}{log_2(i)} \tag{11}$$

where $rel_i$ is the graded relevance of the result at position i.

Let us consider an example for calculating DCG:

Presented with a list of documents in response to a search query, an experiment participant is asked to judge the relevance of each document to the query. Each document is to be judged on a scale of 0-3 with 0 meaning irrelevant, 3 meaning completely relevant, and 1 and 2 meaning "somewhere in between". For the documents ordered by the ranking algorithm as $D_1, D_2, D_3, D_4, D_5, D_6$

the user provides the following relevance scores: 3, 2, 3, 0, 1, 2

That is: document 1 has a relevance of 3, document 2 has a relevance of 2, etc. DCG is used to emphasize highly relevant documents appearing early in the result list. Using the logarithmic scale for reduction, the DCG for each result in order is:

| i | $rel_i$ | $log_2 i$ | $\frac{rel_i}{log_2 i}$ |
|---|---|---|---|
| 1 | 3 | 0 | NA |
| 2 | 2 | 1 | 2 |
| 3 | 3 | 1.585 | 1.892 |
| 4 | 0 | 2.0 | 0 |
| 5 | 1 | 2.322 | 0.431 |
| 6 | 2 | 2.584 | 0.774 |

# 4 Test Collections

To measure ad hoc information retrieval effectiveness in the standard way, we need a test collection consisting of three things:

1. A document collection
2. A test suite of information needs, expressible as queries
3. A set of relevance judgments, standardly a binary assessment of either relevant or non-relevant for each query-document pair.

Across the different definitions of Information Retrieval(IR) from different authors, there is a constant, IR systems have to deal with incomplete or underspecified information in the form of the queries issued by users. The IR systems receiving such queries need to fill in the gaps of the users. For example, Users querying on a web search engine for “BBC” are probably looking for the official home page of the corporation, yet they fully expect the search engine to infer that specific information request from the three letters entered. The fact that the underspecified query content being searched is typically unstructured and its components (i.e., words) can have multiple senses, and different words can be used to express the same concept, merely adds to the challenge of locating relevant items. In contrast to a DB system, whose search outputs are deterministic, the accuracy of an IR system's output cannot be predicted with any confidence prior to a search being conducted; consequently, empirical evaluation has always been a critical component of Information Retrieval.

The vast majority of published IR research assessed effectiveness using a resource known as a test collection used in conjunction with evaluation measures. Such is the importance of test collections that at the time of writing, there are many conferences and meetings devoted purely to their use: including four international conferences, CRANFIELD,TREC, CLEF, and NTCIR, which together have run more than 30 times since the early 1990s.

In the possession of an appropriate test collection, an IR developer or researcher simply loads the documents into their system and in a batch process, submits the topics to the system one-by-one. The list of the docids retrieved for each of the topics is concatenated into a set, known as a run. Then the content of the run is examined to determine which of the documents retrieved were present in the qrels and which were not. Finally, an evaluation measure is used to quantify the effectiveness of that run.

Using test collections, researchers can assess a retrieval system in isolation helping locate points of failure, but more commonly, collections are used to compare the effectiveness of multiple retrieval systems. Either rival systems are compared with each other, or different configurations of the same system are contrasted. Such determinations, by implication, predict how well the retrieval systems will perform relative to each other if they were deployed in the operational setting simulated by the test collection.

## 4.1 Building Test Collections

A test collection usually consists of a document collection, a set of topics that describe a user's information need and a set of relevance judgments indicating which documents in the collection are relevant to each topic. When constructing a test collection there are typically a number of practical issues that must be addressed. By modifying the components of a test collection and evaluation measures used, different retrieval problems and domains can be simulated. The original and most common problem modelled is ad hoc retrieval: the situation in which an information retrieval system is presented with a previously unseen query. However, test collection-based evaluations have also been carried out on tasks including question answering, information filtering, text summarization, topic detection and tracking, image and video retrieval, and text summarization.

*Document collection*: A test collection must contain a static set of documents that should reflect the kinds of documents likely to be found in the operational setting or domain. This might involve digital library collections or sets of Web pages; texts or multimedia items (e.g., images and videos). The notion of a static document collection is important as it ensures that results can be reproduced upon re-use of the test collection.

*Topics*:Information retrieval systems are evaluated for how well they answer users' search requests. In the case of ad hoc retrieval, the test collection must contain a set of statements that describe typical users' information needs. These might be expressed as queries that are submitted to an information retrieval system, questions or longer written descriptions. For example, TREC uses the notion of a topic, which typically consists of three fields: query, title and description. The query field represents a typical set of keywords a user might issue for a given topic. The title field provides a longer description, normally a sentence, of an information need. The description field describes in more detail a user's information and what they are attempting to find. This often includes a description of what constitutes relevant (and non-relevant) documents and may be used by people judging relevance (if not the person who generated the topic).

*Relevance assessments*:For each topic in the test collection, a set of relevance judgments must be created indicating which documents in the collection are relevant to each topic. The notion of relevance used in the Cranfield approach is commonly interpreted as topical relevance: whether a document contains information on the same topic as the query. In addition, relevance is assumed to be consistent across assessors and static across judgments.Relevance judgments can be binary (relevant or not relevant) or use graded relevance judgments, e.g. highly relevant, partially relevant or non-relevant. The use of binary versus graded relevance judgments is important as it has implications for which evaluation measures can be used to evaluate the information retrieval systems. Commonly the assessors will be the people who originally created the topics, but this is not always the case. For example, relevance assessments might be gathered in a distributed way using multiple assessors and crowd sourcing.

There are various ways of gathering the relevance assessments. For example, in TREC the common approach is to gather the top n results from the different information retrieval systems under test for each topic and aggregate results into a single list of results for judging (called pooling). This assumes that the result lists of different information retrieval systems are diverse and therefore will bring relevant documents into the pool. The relevance assessors then go through the pool and make relevance judgments on each document which can then be used to compute system effectiveness. Documents which are not judged are often categorised as not relevant. An issue with pooling is the completeness of relevance assessments. Ideally for each topic one should find all relevant documents in the document collection; however, pooling may only find a subset. Approaches to help overcome this include using results lists from searches conducted manually in the pool of documents for assessment, or supplementing the sets of relevance judgments with additional relevant documents discovered during further manual. Generating complete sets of relevance judgments helps to ensure that when evaluating future systems, improvements in results can be detected. Generating relevance assessment is often highly time-consuming and labor intensive. This often leads to a bottleneck in the creation of test collections. Various techniques have been proposed to make the process of relevance assessment more efficient.

*QREL*: A list of relevant documents (often called qrels) for each query that is required in computing system effectiveness with relevance-based measures (e.g., precision and recall).It contains for each query the set of all documents judged as relevant or non-relevant. The QREL file has the form,

*query-number 0 document-id relevance*

where *query-number* is the number of the query, *document-id* is the external ID for the judged documents, 0 is a constant and relevance is the relevance assigned to the document for the particular query; relevance is either 0 (non-relevant) or 1 (relevant).

*Utility of Data*:The results of query log analysis have many uses in evaluation and tuning.

1. Inferred relevance can produce precision estimates across tens of thousands of users.
2. Similar queries point out different phrasings of the same information need, or similar phrasings for different information needs.
3. Queries that tend to be repeated by the same or different users suggest caching strategies.
4. If a user returns and repeats the same query, you can provide a better ranking based on their prior interaction.

# 5 Significance tests

IR and other experimental sciences are concerned with measuring the effects of competing systems and deciding whether they are really different. For instance, “Does stemming improve my results enough that my search engine should use it?”. Statistical hypothesis testing is a collection of principled methods for setting up these tests and making justified conclusions from their results.
In statistical hypothesis testing, we try to isolate the effect of a single change so we can decide whether it makes an impact. The test allows us to choose between the *null hypothesis* and an *alternative hypothesis.*
*Null Hypothesis*: what we believe by default-the change did not improve performance.
*Alternative Hypothesis*:the change improved performance.

## 5.1 Test Steps and Error Types

The test steps carried out in hypothesis testing are as follows:

1. Prepare your experiment carefully, with only one difference between the two systems: the change whose effect you wish to measure. Choose a significance level , used to make your decision.
2. Run each system many times (e.g. on many different queries), evaluating each run (e.g. with AP).
3. Calculate a test statistic for each system based on the distributions of evaluation metrics.
4. Use a statistical significance test to compare the test statistics (one for each system). This will give you a p-value: the probability of the null hypothesis producing a difference at least this large.
5. If the p-value is less than , reject the null hypothesis.

The probability that you will correctly reject the null hypothesis using a particular statistical test is known as its power.

*Error Types*:Hypothesis testing involves balancing between two types of errors:

- Type I Errors, or false positives, occur when the null hypothesis is true, but you reject it

- Type II Errors, or false negatives, occur when the null hypothesis is false, but you don't reject it.

The probability of a type I error is $\alpha$ −the significance level. The probability of a type II error is $\beta = (1 - power)$.

## 5.2 Popular Significance Tests

In this section we will look at two specific significance tests.

- T-Tests
- Wilcoxon Signed Ranked Test

There are many types of T-Tests, but here we'll focus on two:

- *One-sample tests* have a single distribution of test statistics, and compare its mean to some predetermined value $\mu$.
- *Paired-sample tests* compare the means of two systems on the same queries.

**One Sample T-Tests**: Suppose you were developing a new type of IR system for your company, and your management decided that you can release it if its precision is above 75%.To check this, run your system against 50 queries and record the mean of the precision values. Then calculate the t- value and p-value that correspond to your vector of precision values.
Let $\overline{x}$ :=the mean of values (we assume x is normally distributed)
s := the std. dev of values
n := the number of samples
$\mu$ := the target mean
Then,

$$t = \frac{\overline{x} - \mu}{(s/\sqrt{n})} \tag{12}$$

t is on the student's t-distribution with n-I degrees of freedom.

**Paired-Sample T-tests** Suppose you have runs from two different IR systems: a baseline run using a standard implementation, and a test run using the changes you're testing. You want to know whether your changes outperform the baseline.To test this, run both systems on the same 50 queries using the same document collections and compare the difference in AP values per query.
Let $x_1$ := the baseline values
x$_2$ := $thetestvalues$
$\overline{d} := \overline{x_1 - x_2}$
s$_d$ := $stddev(x_1 - x_2)$
$n := thenumberofsamples$
$Then,$

$$t = \frac{\overline{d}}{(s_d/\sqrt{n})} \tag{13}$$

t is on the student's t-distribution with n-I degrees of freedom.

**Wilcoxon Signed Ranked Test**:The T-tests we used in the previous session assumed your data are normally-distributed. If they're not, the test has less power and you may draw the wrong conclusion. The Wilcoxon Signed Ranks Test is nonparametric: it makes no assumptions about the underlying distribution. It has less power than a T-test when the data is normally distributed, but more power when it isn't.This test is based on comparing the rankings of the data points implied by their evaluation measure (e.g. AP).
*The Signed Rank Test*

1. Produce a vector of the differences between values for each point.

2. Sort the vector by absolute value.
3. Replace the values with their ranks, but keep the signs. (If there are duplicate values, use the mean of the ranks for all values with the appropriate sign).
4. The test statistic is the sum of these signed ranks.

This algorithm produces a discrete distribution that approximates a Normal distribution with mean 0.

*Calculating Z-Ratios*

$$W = \sum_{i=1}^{n} r_i \tag{14}$$

$$\mu_W = 0 \tag{15}$$

$$\sigma_W = \sqrt{\frac{n(n+1)(2n+1)}{6}} \tag{16}$$

$$z = \frac{(W - \mu_W) \pm 0.5}{\sigma_W} = \frac{W - 0.5}{\sigma_W} \tag{17}$$

where $r_i$ are the signed ranks
$\mu_W$ is the mean of the dist. for W
$\sigma_W$ is the std. dev. of the dist. for W

# 6 Manual Assessment

* create your own QREL
  * assessment disagreemnts, fatigue
  * experts vs users vs random people

## 6.1 Crowdsourcing

* cost vs benefit
  * noise
  * quality assurance

# 7 User Studies

* users vs metrics
  * selecting users
  * IRB
  * types of studies
  * types of measurements