INLS 490-154W: Information Retrieval Systems Design and Implementation. Fall 2009.

9. Evaluation-2

Chirag Shah*
School of Information & Library Science (SILS)
UNC Chapel Hill NC 27599
chirag@unc.edu

1 Introduction

Measuring the effectiveness of an IR system is one of the core issues in IR. We already saw some of the most popular measures, which we will review here and then look at some other measures. In addition, we will also talk about how to compare different rank lists.

2 Measures for a single query

For most evaluation measures, a rank list produced by a system as a result of executing a given query is the unit to evaluate the performance of the system. We already looked at measuring the "goodness" of a rank list using recall and recision. Recall is the fraction of relevant documents retrieved. Precision is the fraction of retrieved documents that are relevant. In addition to this, we saw how we can average the precision values at certain point, thus calculating a single number (average precision) to indicate the performance of a given query. Then we saw R-precision, which is the precision after R documents retrieved, where R is the total number of relevant documents for a given query. Average precision and R-precision are shown to be highly correlated.

All of these measures assumed that we have relevance judgments for the collection that we have given the query. However, this will not be the case most times. It is important to have some measure where we can still measure the performance without having all the judgments. A measure called *bpref*, which stands for binary preference, does exactly this (Buckley & Voorhees, 2004). It computes a preference of whether judged relevant documents are retrieved ahead of judged non-relevant documents. It is defined as

$$bpref = \frac{1}{R}\sum_{r}\left(1 - \frac{|n \text{ ranked higher than } r|}{min(R, N)}\right) \tag{1}$$

R: number of judged relevant documents
N: number of judged non-relevant documents
r: relevant document retrieved
n: member of the first R judged non-relevant documents retrieved

As we can see, *bpref* allows us to simply look at how known relevant and non-relevant documents are ranked rather than expecting to know all the relevant documents in the collection.

All the evaluation measures that we have so far are used to evaluate the effectiveness of retrieval through the entire rank list (or whatever part of it that we wish to consider). Sometimes, however, we care about getting only one relevant result. Home-page finding is such a task. In these situations, it does not really matter how well we are doing down the rank list; we simply want to get the relevant result (possibly only one) as high as possible in the rank list. Reciprocal of rank (RR) is a measure that allows us to specifically focus on the rank of a relevant document. It is defined as

$$RR = \frac{1}{rank} \tag{2}$$

Here, *rank* is the rank of the first relevant document in the rank list. The value of RR, as we can see, can range between 0 and 1, with 1 being the best score, when the relevant document is at rank 1.

3 Measures for a system (collection of queries)

While talking about the effectiveness of a system, we need to somehow combine its performance with individual queries. We had seen how MAP (Mean Average Precision) takes average precisions over a number of queries that a system executes and averages them. For the reference, following are the formulations for both average precision (over m recall points) and MAP (over Q queries).

$$AP = \frac{1}{m}\sum_{j=1}^{m} Precision(Recall_j) \tag{3}$$

$$MAP = \frac{1}{|Q|}\sum_{i=1}^{Q} AP_i \tag{4}$$

MAP gives us the arithmetic mean of the average precisions, which is useful in many situations, but this measure does not differentiate between the relative improvements that a system brings to an individual query. Taking geometric precision addresses this issue. The formulation for geometric mean average precision (GMAP) is shown below along with its expansion to more practical form.

$$GMAP = \sqrt[|Q|]{\prod_{i=1}^{Q} AP_i} \tag{5}$$

$$= exp\frac{1}{|Q|}\sum_{i=1}^{Q} logAP_i \tag{6}$$

Now imagine a system improves AP of a query from 0.02 to 0.04. This is not much of an improvement as far as arithmetic mean is concerned, but considering this number in a product, as GMAP does, we can see the difference at a larger scale.

Just as MAP and GMAP extends average precision from a query to system, we can extend rank-based measure of RR as mean reciprocal rank (MRR).

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{Q} \frac{1}{rank_i} \tag{7}$$

4 Evaluations examples

Let us see how various evaluation measures that we saw so far can be computed using the same example as the last time (Figure 3).

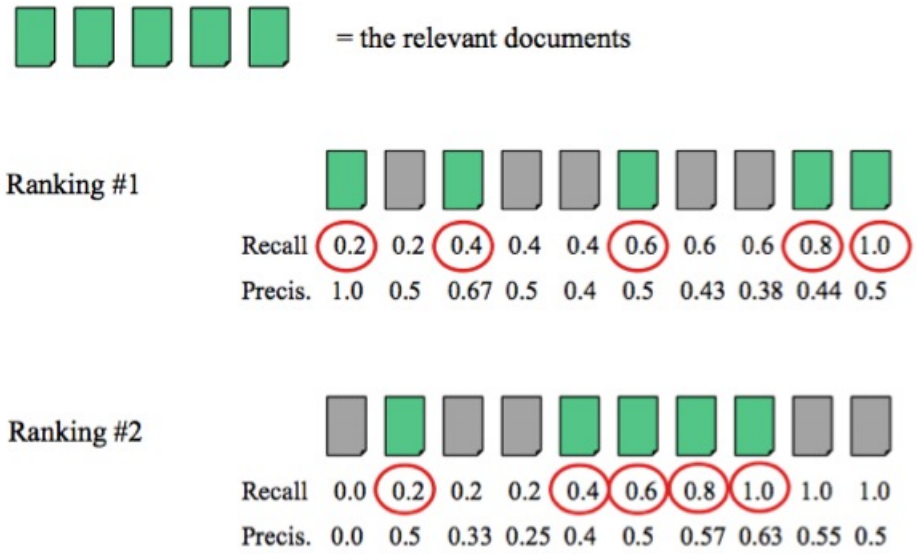

Figure 1: Calculating recall and precision with rank lists (Courtesy: James Allan, UMass Amherst)

As we saw before, AP for these rank-lists are 0.622 and 0.520. If these rank-lists were produced by the same system, then

$$MAP = \frac{0.622 + 0.520}{2} = 0.571 \tag{8}$$

$$GMAP = \sqrt{0.622 * 0.520} = 0.569 \tag{9}$$

RR is the reciprocal of the rank of the first relevant document, which comes out to be 1.0 for the first ranking and 0.5 for the second one. Thus,

$$MRR = \frac{1.0 + 0.5}{2} = 0.75 \tag{10}$$

Let us now calculate *bpref* for both the rank-lists. We know that the total number of judged relevant documents (R) and judges non-relevant documents (N) are 5. Thus, $min(R, N) = 5$. Now, computing $(1 - |n$ ranked higher than $r|/min(R, N))$ at every point when a new retrieved document is judged to be a relevant one, and summing them according to Equation (1),

$$bpref_1 = \frac{1}{5}\left((1 - 0/5) + (1 - 1/5) + (1 - 3/5) + (1 - 5/5) + (1 - 5/5)\right) = 0.44 \tag{11}$$

$$bpref_2 = \frac{1}{5}\left((1 - 1/5) + (1 - 3/5) + (1 - 3/5) + (1 - 3/5) + (1 - 3/5)\right) = 0.48 \tag{12}$$

The example in Figure 3 is probably not very appropriate for showing the usefulness of *bpref* since all the documents are judged. *bpref*, by the nature of its formulation, can handle "gaps" in judgments; it is based on penalizing non-relevant document retrieved before a relevant document, but does nothing to a document that was not judged.

5 Comparing rank lists

All the measures that we saw so far allow us to evaluate the performance of different IR systems and compare them. However, they do not necessarily tell us how significant the difference in performances are. In order to evaluate this difference, we can employ a number of statistical tests, such as Pearson's covariance and Spearman's Rho. One popular test that is more specific to our purpose of comparing rank lists is Kendall's Tau. It is defined as

$$\tau = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \tag{13}$$

n: total number of items
$\frac{1}{2}n(n-1)$: number of pairs
n_c: number of pairs in concordance
n_d: number of pairs in discordance

Once the value of τ is found from the above test, we can map it to z or t distribution for statistical significance. Let us see the calculation of τ with an example.

Consider a set of four documents named a, b, c, and d. System#1's scores are $(a = 0.4, b = 0.3, c = 0.2, d = 0.1)$, thus ranking them as $\{a, b, c, d\}$. System#2's scores are $(a = 0.4, b = 0.1, c = 0.25, d = 0.05)$, ranks them as $\{a, c, b, d\}$. The possible pairs with 4

items for any given system maintaining the order is 6. Of these six pairs, we can see that items b and c are ordered differently by each system. Thus, we have $n_c = 5$ and $n_d = 1$. Substituting these values in Equation (8), we get $\tau = 0.67$.

Let us see how we can use statistical package R[1] to compute this. Let us input the document scores for both the systems. This is done by the assignment operation (Code 1).

Once we assign the scores to x and y, we can ask R to rank them. This is done using `rank` function. However, by default `rank` organizes the objects in increasing order of their values (here, scores). We need the ranks in decreasing order of the scores. Therefore, we can use `order(-x)` to rank x. Similarly we can rank the documents represented by y using `order(-y)`.

R has a function called `cor`, which can find correlation between two lists or distributions. Here, our two rank lists are represented by `order(-x)` and `order(-y)`. Code-1 shows how different kinds of correlation can be easily computed with this function. As we can see, Kendall's Tau calculation gives us the same value that we found before.

Code 1: Commands and output for R to compare two rank lists

```
> x <- c(0.5,0.4,0.3,0.2)
> y <- c(0.4,0.1,0.25,0.05)
> order(-x)
[1] 1 2 3 4
> order(-y)
[1] 1 3 2 4
> cor(order(-x), order(-y), method="pearson")
[1] 0.8
> cor(order(-x), order(-y), method="spearman")
[1] 0.8
> cor(order(-x), order(-y), method="kendall")
[1] 0.6666667
```

6 Summary

- Performance of a query can be measured using recall, precision, average precision, R-precision, bpref, and reciprocal rank.
- Performance of a system can be measured using MAP, GMAP, and MRR.
- There is no "perfect" evaluation measure. Choosing "right" measure(s) to evaluate an IR system depends on the task and requirements.

References

Buckley, C., & Voorhees, E. M. (2004, July 25–29). Retrieval evaluation with incomplete information. In *Proceedings of ACM SIGIR.* Sheffield, South Yorkshire, UK.

[1] http://www.r-project.org/