The Evaluation Tool in LETOR

Jun Xu, Tie-Yan Liu, and Hang Li
Microsoft Research Asia
letor@microsoft.com

1. Introduction

Together with the benchmark datasets, we also release a set of evaluation tools. The tools were written in perl, and can output precision at position n (P@n), mean average precision (MAP), normalized discount cumulative gain (NDCG) for the ranking results of a given ranking algorithm, and the significance test results for two given ranking algorithms. By using a single set of evaluation tools, the experimental results of different methods can be easily and impartially compared.
In this document, we describe the usage of the tools, as well as the evaluation metrics they supports.

2. Command Line

We have three tools in the package.

1) Eval-Score.pl

```
perl  Eval-Score.pl  [test file] [prediction file] [output file] [flag]
```

There are four arguments for the command of the tool:

[test file]:
The input file of the evaluation tool. For example, "OHSUMED\Data\Fold*n*\testset.txt". Actually it contains the relevance (label) information of the testset and is used for evaluation.

[prediction file]:
The input file of the evaluation tool, containing the scores given by the ranking algorithm under investigation. Note that each score occupies one line, and corresponds to one document in the testset.

[output file]:
The output file of the evaluation tool, containing the evaluation results of P@n, MAP and NDCG. The definitions of these measures will be given in the next section.

[flag]:

If flag is '1', the evaluation tool will output per-query evaluation results; otherwise it will output the average evaluation results.

2) Eval-Rank.pl

```
perl  Eval-Rank.pl  [test file] [prediction file] [output file] [flag]
```

Considering that if we use prediction scores as input, the evaluation results may depend on how we deal with ties (documents with exactly the same scores). To avoid this ambiguity, we provide another tool which uses ranked position as input. Users can use their own heuristics to handle ties, and evaluate the corresponding results with this tool.

There are also four arguments for the command of the tool:

[test file]:
The input file of the evaluation tool. For example, “OHSUMED\Data\Fold*n*\testset.txt”. Actually it contains the relevance (label) information of the testset and is used for evaluation.

[prediction file]:
The input file of the evaluation tool, containing the rank positions given by the ranking algorithm under investigation. Note that each rank position occupies one line, and corresponds to one document in the testset.

[output file]:
The output file of the evaluation tool, containing the evaluation results of P@n, MAP and NDCG.

[flag]:

If flag is ‘1’, the evaluation tool will output per-query evaluation results; otherwise it will output the average evaluation results.

2) LETOR-ttest.pl

```
Perl Eval-ttest.pl [result file A] [result file B] [output]
```

There are three arguments for the command of the tool:

[result file A]:
The input file of the evaluation tool. It is the result file output by the Eval-Rank.pl or Eval-Score.pl for a specific ranking algorithm, with the flag equal to 1 (query-level evaluation results).

[result file A]:
The input file of the evaluation tool. It is the result file output by the Eval-Rank.pl or Eval-Score.pl for another ranking algorithm, with the flag equal to 1 (query-level evaluation results).

[output file]:
The output file of the evaluation tool, containing the t-test result for P@n, MAP and NDCG between these two algorithms.

Note that, to use the significance test tool, one need install the following perl packages:

- Statistics::DependantTTest;
- Statistics::Distributions;
- Statistics::PointEstimation;
- Statistics::Descriptive.

These packages can be downloaded from http://www.cpan.org/modules/by-module/Statistics/.

3. Evaluation Metrics

As aforementioned, the evaluation tool can output P@N, MAP, NDCG and NDCG2 for a given ranking algorithm. We will describe the definitions of these four metrics in this section.

Precision at position n (P@n)

Precision at *n* measures the relevance of the top *n* results of the ranking list with respect to a given query.

$$P@n = \frac{\#\, relevant\ docs\ in\ top\ n\ results}{n}$$

For example, if the top 10 documents returned for a query are {*relevant, irrelevant, irrelevant, relevant, relevant, relevant, irrelevant, irrelevant, relevant, relevant*}, then P@1 to P@10 values will be {1, 1/2, 1/3, 2/4, 3/5, 4/6, 4/7, 4/8, 5/9, 6/10} respectively. For a set of queries, one averages the P@*n* values of all the queries to get the mean P@*n* value.

Mean average precision (MAP)

For a single query, average precision is defined as the average of the P@*n* values for all relevant documents.

$$AP = \frac{\sum_{n=1}^{N}(P@n * rel(n))}{\#total\ relevant\ \ docs\ for\ this\ query}$$

where *N* is the number of retrieved documents, and *rel*(*n*) is a binary function on the relevance of the *n*-th document.

$$rel(n) = \begin{cases} 1, if\ the\ n^{th}\ doc\ is\ relevant \\ 0, otherwise \end{cases}$$

Similar to mean P@*n*, over a set of queries, we get MAP by averaging the AP values of all the queries.

Normalized discount cumulative gain (NDCG)

Note that P@*n* and MAP can only handle cases with binary judgment: “relevant” or “irrelevant”. Recently, a new evaluation metric called Normalized Discount Cumulative Gain (NDCG) has been proposed, which can handle multiple levels of relevance. While evaluating a ranking list, NDCG follows two rules:

1) Highly relevant documents are more valuable than marginally relevant document;
2) The lower ranking position a document (of any relevance level) has, the less valuable it is for the user, because it is less likely to be examined by the user.

According to the above rules, there are four steps to compute the NDCG value for a ranking list:

1) Compute the gain of each document;
2) Discount the gain of each document by its ranking position;
3) Cumulate the discounted gain of the list;
4) Normalize the discounted cumulative gain of the list.

NDCG of a ranking list at position *i* is calculated as follow according to the original paper [1][2]:

$$N(n) \equiv Z_n \sum_{j=1}^{n} \begin{cases} 2^{r(j)} - 1, j = 1 \\ \dfrac{2^{r(j)} - 1}{\log(j)}, j > 1 \end{cases}$$

where $r(j)$ is the rating of the *j*-th document in the list, and the normalization constant Z_n is chosen so that the perfect list gets a NDCG score of 1.

Note that for the TREC Datasets, there are two relevance levels {0, 1}; and for the OHSUMED Dataset, there are three relevance levels {0, 1, 2}.

4. Additional Note

Users of this evaluation tool need to sign the license agreement provided at the download site, when they download it. For questions or requests, please send email to letor@microsoft.com.

References

[1] Jarvelin, K., and Kekalainen, J. IR evaluation methods for retrieving highly relevant documents. Proceedings of SIGIR 2000, pp.41-48, 2000.

[2] Jarvelin, K., and Kekalainen, J. Cumulated gain-based evaluation of IR techniques, ACM Transactions on Information Systems, 2002.