TREC_EVAL: IR Evaluation

Draft: 10/26/2010

1. **Module name**: Evaluation in Information Retrieval

2. **Scope**: This module addresses the methods used to evaluate an Information Retrieval system. We focus on evaluating a system using relevance and apply the knowledge by using TREC_EVAL.

3. **Learning objectives:**

   Students should be able to:

   - Explain common evaluation techniques for Information Retrieval systems and their uses
   - Evaluate an IR system using TREC_EVAL and interpret the results

4. **5S characteristics of the module (streams, structures, spaces, scenarios, societies)**:

   Four S's are present – Streams, Societies, Scenarios and Structures. The Space component was not considered in this module.

   a. Stream: TREC_EVAL is designed for evaluation of various information retrieval systems. It handles streams of documents, queries, and relevance judgments. Each is made up of a sequence of characters.
   b. Structures: TREC_EVAL has its own architecture that has been derived from the TREC conference that has been widely accepted. Since XML is used, there is structure based on use of tags, identifying the organization of each XML file (according to a schema). Internally, many data structures are used during computation and for reporting.
   c. Scenarios: TREC_EVAL and IR systems interact with each other following a series of steps to achieve tasks. The main one of these involves processing a set of files resulting from an IR experiment.
   d. Societies: This module is used by those in academia for research and teaching, and by various types of IR system evaluators.

5. **Level of effort required (in-class and out-of-class time required for students)**:

   a. In class: Listening to 20 minute long presentation
   b. Outside of class:
      - 2 -3 hours for reading
      - Approximately 1 hour for exercises

6. **Relationships with other modules (flow between modules)**:

   TREC_EVAL is designed for evaluating the information retrieval of a specific

IR system or program. This program has strong associations to many IR systems that are presented in modules. The modules that are most closely associated with this module are Apache SOLR and WordNet. The IR systems that are actively implementing TREC_EVAL are Lemur and Weka.

7. **Prerequisite knowledge/skills required**: Knowledge of alternatives of IR system design. Knowledge of key concepts related to IR system evaluation. Skill to run experiments to determine which techniques are the most effective for use in which applications. Basic skill to use Unix systems and graph drawing tools.

8. **Introductory remedial instruction**:
    a. **IR system evaluation**:

    Can be evaluated using quantitative measures such as:

    - How fast does it index
    - How fast does it search
    - Expressiveness of query language

    A key measure is user happiness

    - What is this?
    - Speed of response/size of index are factors.
    - But blindingly fast, useless answers won't make a user happy.

    Need a way of quantifying user happiness

    b. **Measuring user happiness**

    Issue: who is the user we are trying to make happy? Depends on the setting:

    1. Web engine: user finds what they want and returns to the engine.

    Can measure rate of return of users

    2. Commerce site: user finds what they want and makes a purchase.

    Measure time to purchase, or fraction of searchers who become buyers?

    3. Enterprise (company/govt/academic): Care about "user productivity".

    How much time do my users save when looking for information?

9. **Body of knowledge**
    a. **Relevance of search results**

    How do you measure relevance? Use a test collection.

    A test collection is made of:

    - Document collection
    - Test suite of **information needs**
    - Relevance judgments

    Document is classified as relevant or not.

    Test collection must be of reasonable size.

Relevance is assessed relative to the **information need** *not* the **query.**

E.g., Information need: *I'm looking for information on whether drinking red wine is more effective than white wine at reducing your risk of heart attacks*

Query: ***wine red white heart attack effective reduce risk***

You evaluate whether the document addresses the information need, not whether it has these words.

There are standard test collections available such as: TREC, CLEF, GOV2, NTCIR

b. Unranked retrieval evaluation

Using 2 measures: recall and precision.

- **Precision**: fraction of retrieved docs that are relevant = P(relevant|retrieved)
- **Recall**: fraction of relevant docs that are retrieved = P(retrieved|relevant)

Can use ACCURACY, where given a query, an engine classifies each doc as "Relevant" or "Nonrelevant". The **accuracy** of an engine: the fraction of these classifications that are correct

Accuracy is a commonly used evaluation measure in machine learning classification work.

Why is this not a very useful evaluation measure in IR?

- Can build a 99.9999% accurate search engine: one that displays no results!
- People doing information retrieval *want to find something* and have a certain tolerance for junk.

Can use a combined measure of precision and recall. One that assesses precision/recall tradeoff is **F measure** (weighted harmonic mean):

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

c. Ranked retrieval evaluation

Can evaluate ranked results by using a *precision-recall curve*. To remove jiggles use interpolated precision

To summarize the precision-recall curve one can use:

i. 11-point interpolated average precision

The standard measure in the early TREC competitions: you take the precision at 11 levels of recall, varying from 0 to 1 by tenths, of the documents, using interpolation (the value for 0 is always interpolated!), and average them

ii. Mean average precision (MAP)

Average of the precision value obtained for the top *k* documents, each time a relevant doc is retrieved. Avoids interpolation or use of fixed

recall levels

iii. Precision-at-*k*: Precision of top *k* results

Perhaps appropriate for most of web search: all people want are good matches on the first one or two results pages

But: averages badly and has an arbitrary parameter of *k*

iv. R-precision

have known (though perhaps incomplete) set of relevant documents of size *Rel*, then calculate precision of top *Rel* docs returned. Perfect system could score 1.0.

v. Other measures such as ROC curve and NDCGG

d. Relevance assessments

Test information needs should be germane to the documents in the test document collection

- Need to use human beings! Are human panels perfect?
- Collect relevance assessments, feasible for tiny collections but must use **pooling** in large collections.
- Use kappa statistic as a measure for agreement

e. Kappa measure

Measures

- Agreement measure among judges
- Designed for categorical judgments
- Corrects for chance agreement

Kappa = [P(A) – P(E)] / [1 – P(E)]

- P(A) = proportion of time judges agree
- P(E) = what agreement would be by chance
- Kappa = 0 for chance agreement, 1 for total agreement.

f. Refining IR

An IR system can be modified by deploying variants of the system and recording user satisfaction.

i. A/B testing

- Purpose: Test a single innovation
- Prerequisite: You have a large search engine up and running.
- Have most users use old system
- Divert a small proportion of traffic (e.g., 1%) to the new system that includes the innovation
- Evaluate with an “automatic” measure like clickthrough on first result
- Now we can directly see if the innovation does improve user

happiness.

g. Result summaries

i. The title is typically automatically extracted from document metadata. What about the summaries?

ii. This description is crucial. Users may identify good/relevant hits based on this description.

iii. Two basic kinds:

- A **static summary** of a document is always the same, regardless of the query that led to retrieval of that document.
- A **dynamic summary** is a *query-dependent* attempt to explain why the document was retrieved for the query at hand.

h. Background of TREC_EVAL origins

i. TREC: Text REtrieval Conference (TREC)(http://trec.nist.gov/) Originated from the TIPSTER program sponsored by the Defense Advanced Research Projects Agency (DARPA).

- Became an annual conference in 1992, co-sponsored by the National Institute of Standards and Technology (NIST) and DARPA
- Participants are given parts of a standard set of documents and TOPICS (from which queries have to be derived) in different stages for training and testing.
- Participants submit the P/R values for the final document and query corpus and present their results at the conference.
- This led to creation of evaluation software for the purpose of evaluating the performance of various information retrieval systems on these documents and query results.

ii. TREC_EVAL purpose for TREC

- Provides a common ground for comparing different IR techniques
- Sharing of resources and experiences in developing the benchmark
- Encourage participation from industry and academia
- Development of new evaluation techniques, particularly for new applications

10. Resources

Required readings:

- Chapter 8 of **Introduction to information Retrieval**; *Manning, Raghavan and Schütze*.
- Notes on TREC_EVAL, http://ir.iit.edu/~dagr/cs529/files/project_files/trec_eval_desc.htm

11. Exercises / Learning activities

(The following exercise has been adapted, and data sets used for the exercise have been derived, from http://www.ccs.neu.edu/home/ekanou/ISU535.09X2/Homeworks/hw.01.ekanou.html . The information about Trec_eval was made possible by Notes on TREC_EVAL, given under Resources.

1. In this exercise, you will gain an understanding of how to run trec_eval and understand results gained from it. The input to your program will be two files: (a) the ranked list of documents as returned by a retrieval system, and (b) the qrel file that contains for each query the set of all documents judged as relevant or non-relevant.

The results file has the form,

query-number Q0 document-id rank score Exp

where *query-number* is the number of the query, *document-id* is the external ID for the retrieved document, and *score* is the score that the retrieval system creates for that document against that query. *Q0* (Q zero) and *Exp* are constants that are used by some evaluation software. You can see such a file for the READWARE retrieval system submitted to TREC 8 by accessing the first team 5 cloud instance, IBMcloudTeam5a.

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). You can see such a file for the READWARE retrieval system submitted to TREC 8 by accessing the first team 5 cloud instance, IBMcloudTeam5a.

The format for running the trec_eval program on the command line of the Linux based system is

./ *trec eval* [-q] [-a] trec_qrel_file trec_results_file,

where

- *trec eval* is the executable name for the code
- -q is a parameter specifying detail for all queries
- -a is a parameter specifying summary output only
- trec_qrel _file is the qrels, query relevance file
- trec_results_file is the result file for an IR system.

The results obtained by running trec_eval (summary output only) are

num_ret	*Total number of documents retrieved over all queries*
num_rel	*Total number of relevant documents over all queries*
num_rel_ret	*Total number of relevant documents retrieved over all queries*
map	*Mean Average Precision (MAP)*
gm_ap	*Average Precision. Geometric Mean, q_score=log(MAX(map,.00001))*
R-prec	*R-Precision (Precision after R (= num-rel for topic) documents retrieved)*
bpref	*Binary Preference, top R judged nonrel*
recip_rank	*Reciprical rank of top relevant document*
ircl_prn.0.00	*Interpolated Recall - Precision Averages at 0.00 recall*
ircl_prn.0.10	*Interpolated Recall - Precision Averages at 0.10 recall*
ircl_prn.0.20	*Interpolated Recall - Precision Averages at 0.20 recall*
ircl_prn.0.30	*Interpolated Recall - Precision Averages at 0.30 recall*
ircl_prn.0.40	*Interpolated Recall - Precision Averages at 0.40 recall*
ircl_prn.0.50	*Interpolated Recall - Precision Averages at 0.50 recall*
ircl_prn.0.60	*Interpolated Recall - Precision Averages at 0.60 recall*
ircl_prn.0.70	*Interpolated Recall - Precision Averages at 0.70 recall*
ircl_prn.0.80	*Interpolated Recall - Precision Averages at 0.80 recall*
ircl_prn.0.90	*Interpolated Recall - Precision Averages at 0.90 recall*
ircl_prn.1.00	*Interpolated Recall - Precision Averages at 1.00 recall*
P5	*Precision after 5 docs retrieved*
P10	*Precision after 10 docs retrieved*
P15	*Precision after 15 docs retrieved*
P20	*Precision after 20 docs retrieved*
P30	*Precision after 30 docs retrieved*
P100	*Precision after 100 docs retrieved*
P200	*Precision after 200 docs retrieved*
P500	*Precision after 500 docs retrieved*
P1000	*Precision after 1000 docs retrieved*

It is possible to obtain more output information with inclusion of [-q], which provides query by query summary information. More information can be found by accessing the following link:

http://www-nlpir.nist.gov/projects/trecvid/trecvid.tools/trec_eval_video/README

Compute the MeanAverage Precision of the ranked lists of documents returned by READWARE for all 50 queries. For this, just submit the summary statistics table, recall level precision averages table, and document level averages table. Also provide average precision values for queries 401 and 402.

Using the table obtained also please provide a recall-precision graph and average precision histogram, only for summary output, not for each query. A sample output is provided below for your convenience.

Sample output

The following figures and tables are from the source:
http://trec.nist.gov/pubs/trec15/appendices/CE.MEASURES06.pdf

Table 1: Sample "Summary Statistics" Table.

Summary Statistics	
Run	Cor7A1clt–automatic, title
Number of Topics	50
Total number of documents over all topics	
Retrieved:	50000
Relevant:	4674
Rel_ret:	2621

Table 2: Sample "Recall Level Precision Averages" Table.

Recall Level Precision Averages	
Recall	Precision
0.00	0.6169
0.10	0.4517
0.20	0.3938
0.30	0.3243
0.40	0.2715
0.50	0.2224
0.60	0.1642
0.70	0.1342
0.80	0.0904
0.90	0.0472
1.00	0.0031
Average precision over all relevant docs	
non-interpolated	0.2329

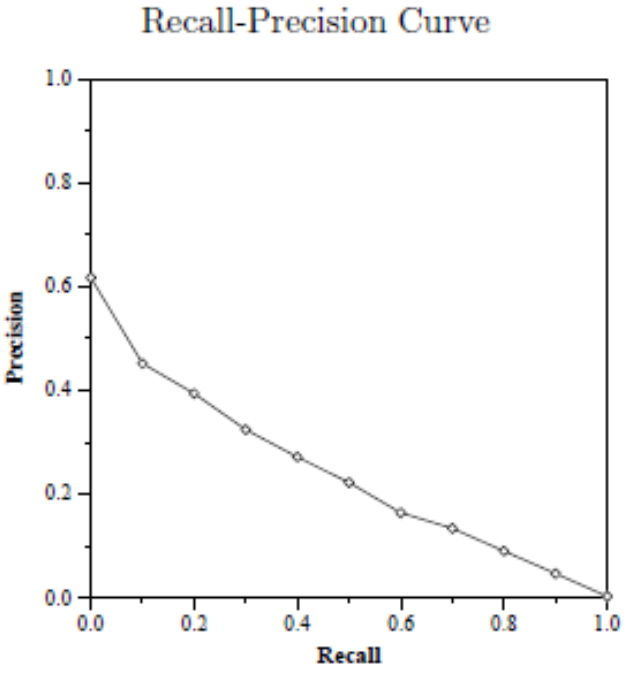

Figure 1: Sample Recall-Precision Graph.

Table 3: Sample "Document Level Averages" Table.

Document Level Averages	
	Precision
At 5 docs	0.4280
At 10 docs	0.3960
At 15 docs	0.3493
At 20 docs	0.3370
At 30 docs	0.3100
At 100 docs	0.2106
At 200 docs	0.1544
At 500 docs	0.0875
At 1000 docs	0.0524
R−Precision (precision after R docs retrieved (where R is the number of relevant documents))	
Exact	0.2564

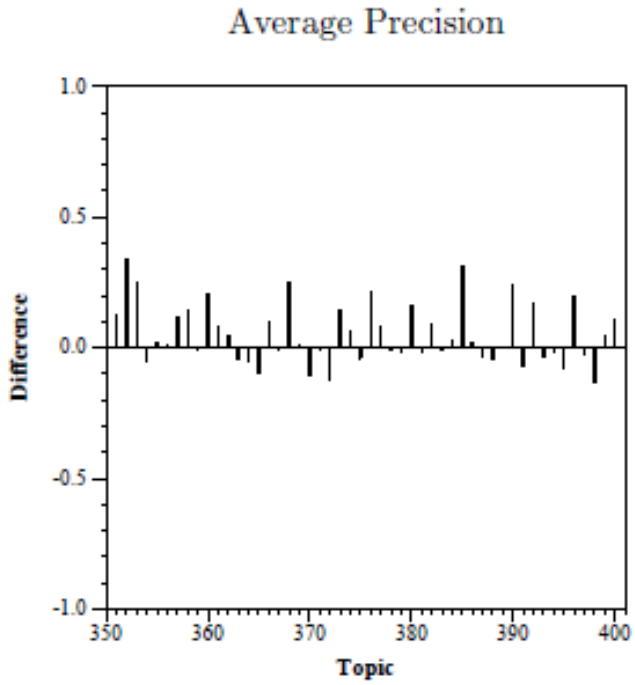

Figure 2: Sample Average Precision Histogram.

Using putty you will be able to access Team 5's cloud instance. We already installed trec_eval for you, which is found in directory **trec_eval.8.1** and the

qrel file and **top(results file)** are found in the directory named **exercise.**

12. Evaluation of learning objective achievement:

In their reports, the students should show good understating of evaluation of an IR system, and of the basics of TREC_EVAL.

13. Glossary

- **Gold Standard**: The result of a team of experts working with a set of queries and documents, classifying each document as relevant or not.
- **Precision**: fraction of retrieved docs that are relevant = P(relevant|retrieved)
- **Recall**: fraction of relevant docs that are retrieved = P(retrieved|relevant)
- **Accuracy** of an engine: the fraction of relevant/nonrelevant classifications that are correct.
- **F measure** (weighted harmonic mean): a combined measure that assesses according to a precision/recall tradeoff.

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

- **Kappa Measure**= [P(A) – P(E)] / [1 – P(E)] (where P(A) = proportion of the time judges agree, P(E) = what agreement would be by chance)
- **Static summary** of a document is always the same, regardless of the query that led to retrieval of that document.
- **Dynamic summary** is a *query-dependent* attempt to explain why the document was retrieved for the query at hand.

14. Additional useful links

- Download trec_eval from

  http://trec.nist.gov/trec_eval/

15. Contributors

a. Initial authors: Bhanu Peddi, Huijun Xiong, Noha ElSherbiny – graduate students in Dept. of Computer Science, Virginia Tech

b. Evaluator: Dr. Edward A. Fox