

How to evaluate recommender algorithms offline based on simulated streams? - A tutorial for replaying the last weeks recorded in CLEF NewsREEL.

Motivation

Everyone who participated in the online task is invited to evaluate the implemented algorithms offline. The idea is to replay the data recorded in the online evaluation. We provide all required components and the log files. No additional implementation is needed. The offline evaluation gives additional insights and helps do analyze the performance of recommender algorithms in detail.

How to setup the task 2 evaluation environment?

The offline evaluation components are implemented in Java. You need a Java SDK (version >1.6) and maven2 (we use version 3.0.3).

Step 1: Build the Component for Creating the Message Stream.

Please execute the following steps:

- Check out the example code: `git clone https://github.com/andreas-dai/NewsREEL-Template` (alternatively, the code can be downloaded from http://data.dai-labor.de/corpus/clef-newsreel-2015/CLEF-May2016/20160327_NewsREEL.zip)
- Build the project: `mvn clean install assembly:assembly`

Step 2: Download the log files to be replayed.

Please execute the following steps:

- Create a new empty directory locally.
- Download the data from files (*.gz) the webserver and store the files in the empty directory <http://data.dai-labor.de/corpus/clef-newsreel-2015/CLEF-May2016/>
It is not recommended to gunzip the data files - the provided components can handle gz files.
- Note: Participants have reported that gz files get corrupted while downloading the data files. Please use an alternative browser in this case.

Step 3: Simulate the server that sends the recommendation requests:

Please execute the following steps:

- Determine the URL of the recommender you would like to evaluation. Typically, you find the address in the ORP dashboard.
- Ensure that the online evaluation is disabled for the recommender - a restart of the recommender typically ensures that an older online evaluation does not influence the offline evaluation.
- Change to the directory containing the built java code. Typically this is `20160327_NewsREEL/target`. The directory should contain the file "clef-newsreel-challenge-0.0.1-jar-with-dependencies.jar"
- Start the offline evaluation with the following command:
`java -cp clef-newsreel-challenge-0.0.1-jar-with-dependencies.jar`

eu.crowdrec.contest.sender.RequestSenderORP <URL of your recommender> <the directory containing the data gz files> <the filename for the output file>

Notes:

- The <URL of your recommender> typically looks like: http://XXX.XXX.XXX.XXX:80
- The file "<the directory containing the data gz files>" can be relative or absolute. Instead of a directory you can also provide a gz-file instead.
- The file "<the filename for the output file>" is automatically created. It stores all answered from the recommender for the requests.
- The evaluation may require a lot of time. In order to check the current state, you can check the last line of the output file. The 3rd token contains the timestamp of most recently processed request (note: you have to remove the last 3 digits when using an Unix timestamp converter)

Step 4: Evaluate the recommendation precision

Finally, the offline CTR must be computed. The offline CTR is computed based on the output file (created in the previous step) and the data files (*.gz).

- Execute the following command:
java -Xmx1G -cp clef-newsreel-challenge-0.0.1-jar-with-dependencies.jar
eu.crowdrec.contest.evaluation.Evaluator <outputfile from step 3> <the directory containing the data gz files> 600000 t

Notes:

- The parameter 600000 means that a recommendation is correct, if the user accesses the article within the next 10 minutes (600,000 ms).
- The CTR is written on stdout. You can redirect the output to a file by appending "> CTR-outfile.txt" to the command.
- In addition to the CTR a histogram for the response time distribution is created.
- The Evaluation implicitly creates a statistic on an hourly basis. The data are stored in a new file in the current directory. The files starts with "detailedStatistic" and ends with the name of the given output file.