

How to evaluate recommender algorithms based on offline simulated streams using Idomaar? - A tutorial for CLEF NewsREEL Task 2

Motivation

The CLEF NewsREEL task 2 focusses on the offline evaluation of stream recommendation algorithms. This allows researches analyzing the characteristics of algorithm. In contrast to the live evaluation in task1 the experiments are reproducible by replaying the provided data stream several times. In addition, the debugging of the offline streams can be done without time-constraints simplifying the identification of bottlenecks or deadlocks.

How to setup the task 2 evaluation environment?

We show step by step how to setup an evaluation environment for task 2. Here we explain a first setup focusing on benchmarking the recommendation quality. We consider the technical complexity by using an exactly defined computing environment ("virtual box") and the time needed for processing the dataset. An extended resource measurement will be added soon.

In the tutorial we use the example project providing all components for an evaluation run. It is recommended to check out the example project and to retrace the explained steps.

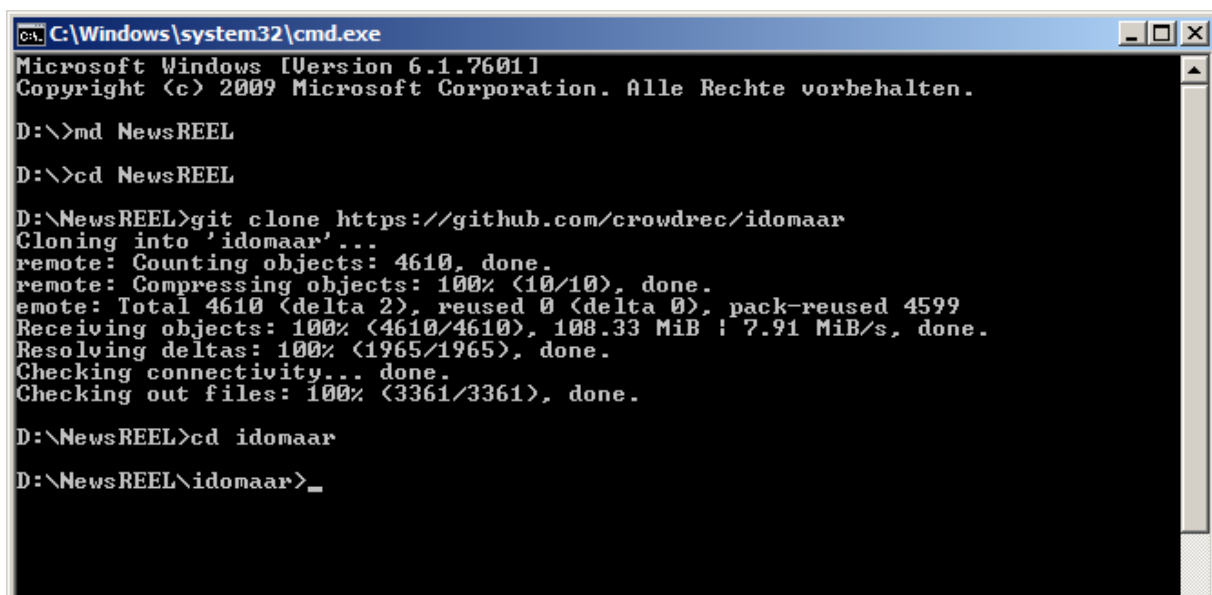
Step 1: Prepare the evaluation environment

Ensure that you have installed the following components.

- A git client for checking out the project from gitHub
- Vagrant
- Virtual box (supporting 64-bit OS)

Checkout Idomaar from GitHub:

```
$ git clone https://github.com/crowdrec/idomaar
```



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Alle Rechte vorbehalten.

D:\>md NewsREEL

D:\>cd NewsREEL

D:\NewsREEL>git clone https://github.com/crowdrec/idomaar
Cloning into 'idomaar'...
remote: Counting objects: 4610, done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 4610 (delta 2), reused 0 (delta 0), pack-reused 4599
Receiving objects: 100% (4610/4610), 108.33 MiB | 7.91 MiB/s, done.
Resolving deltas: 100% (1965/1965), done.
Checking connectivity... done.
Checking out files: 100% (3361/3361), done.

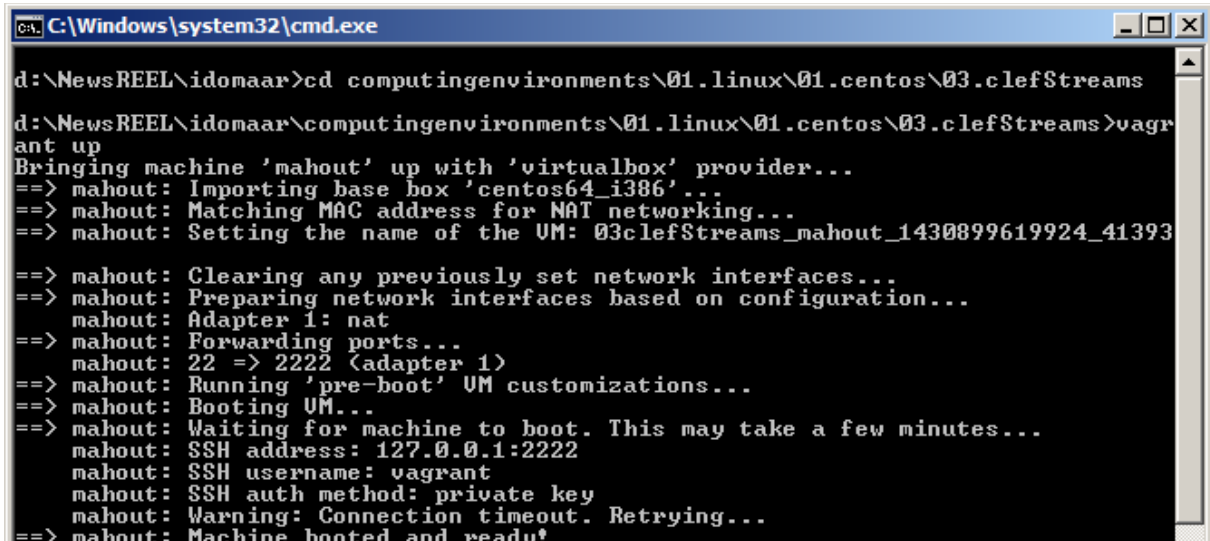
D:\NewsREEL>cd idomaar

D:\NewsREEL\idomaar>_
```

Step 2: Start a virtual machine

Select one of the pre-defined virtual computing environments and start the virtual machine using vagrant:

```
$ cd computingenvironments\01.linux\01.centos\03.clefStreams
$ vagrant up
```



```
C:\Windows\system32\cmd.exe
d:\NewsREEL\idomaar>cd computingenvironments\01.linux\01.centos\03.clefStreams
d:\NewsREEL\idomaar\computingenvironments\01.linux\01.centos\03.clefStreams>vagr
ant up
Bringing machine 'mahout' up with 'virtualbox' provider...
==> mahout: Importing base box 'centos64_i386'...
==> mahout: Matching MAC address for NAT networking...
==> mahout: Setting the name of the UM: 03clefStreams_mahout_1430899619924_41393
==> mahout: Clearing any previously set network interfaces...
==> mahout: Preparing network interfaces based on configuration...
mahout: Adapter 1: nat
==> mahout: Forwarding ports...
mahout: 22 => 2222 (adapter 1)
==> mahout: Running 'pre-boot' UM customizations...
==> mahout: Booting UM...
==> mahout: Waiting for machine to boot. This may take a few minutes...
mahout: SSH address: 127.0.0.1:2222
mahout: SSH username: vagrant
mahout: SSH auth method: private key
mahout: Warning: Connection timeout. Retrying...
==> mahout: Machine booted and ready!
```

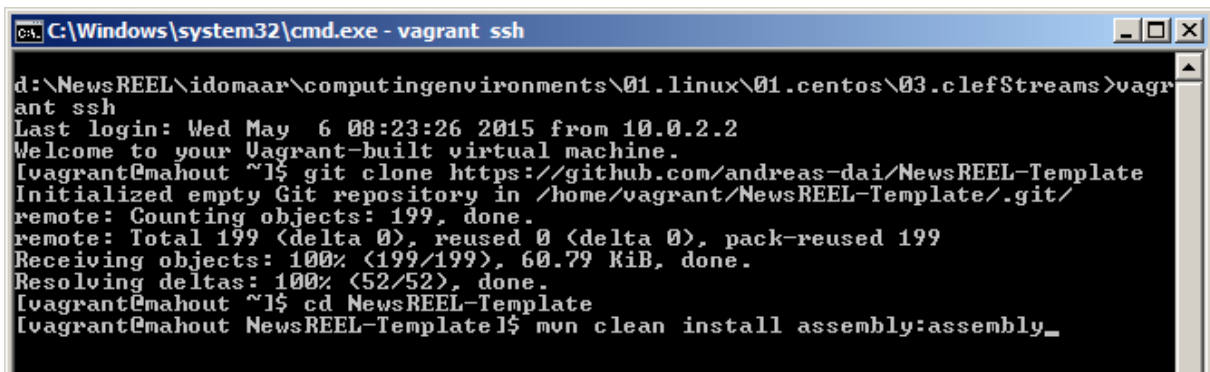
Step 3: Start a recommender algorithm inside the virtual machine

Login to the started virtual machine.

```
$ vagrant ssh
```

Clone the example recommender from gitHub inside the virtual machine. Compile the java code and build a jar file (note that the virtual machine runs Linux)

```
[~]$ git clone https://github.com/andreas-dai/NewsREEL-Template
[~]$ cd NewsREEL-Template
[~]$ mvn clean install assembly:assembly
```



```
C:\Windows\system32\cmd.exe - vagrant ssh
d:\NewsREEL\idomaar\computingenvironments\01.linux\01.centos\03.clefStreams>vagr
ant ssh
Last login: Wed May 6 08:23:26 2015 from 10.0.2.2
Welcome to your Vagrant-built virtual machine.
[vagrant@mahout ~]$ git clone https://github.com/andreas-dai/NewsREEL-Template
Initialized empty Git repository in /home/vagrant/NewsREEL-Template/.git/
remote: Counting objects: 199, done.
remote: Total 199 (delta 0), reused 0 (delta 0), pack-reused 199
Receiving objects: 100% (199/199), 60.79 KiB, done.
Resolving deltas: 100% (52/52), done.
[vagrant@mahout ~]$ cd NewsREEL-Template
[vagrant@mahout NewsREEL-Template]$ mvn clean install assembly:assembly_
```

Start the recommender.

```
[~]$ java -cp ./target/clef-newsreel-challenge-0.0.1-jar-with-dependencies.jar de.dailab.plistacontest.client.Client 0.0.0.0:8088 log4j.properties
```

The command starts the class "Client" contained in the jar file build in the previous step. The program expects two parameters. The first parameter defines the address in that the client waits for messages. We use the IP address 0.0.0.0 ("all network interfaces") and the port 8088. The second parameter defined the logging configuration.

```
[INFO] META-INF/maven/ already added, skipping
[INFO] org/ already added, skipping
[INFO] org/apache/ already added, skipping
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 24.529 s
[INFO] Finished at: 2015-05-06T08:31:00+00:00
[INFO] Final Memory: 13M/32M
[INFO] -----
[vagrant@mahout NewsREEL-Template] $ java -cp ./target/clef-newsreel-challenge-0.0.1-jar-with-dependencies.jar de.dailab.plistacontest.client.Client 0.0.0.0:8088 log4j.properties
```

Note: The virtual machine is running Linux. Thus the "path separator" is a colon ":".

Step4: Build the NewsREEL evaluator

Start the NewsREEL evaluator virtual machine and build the evaluator.

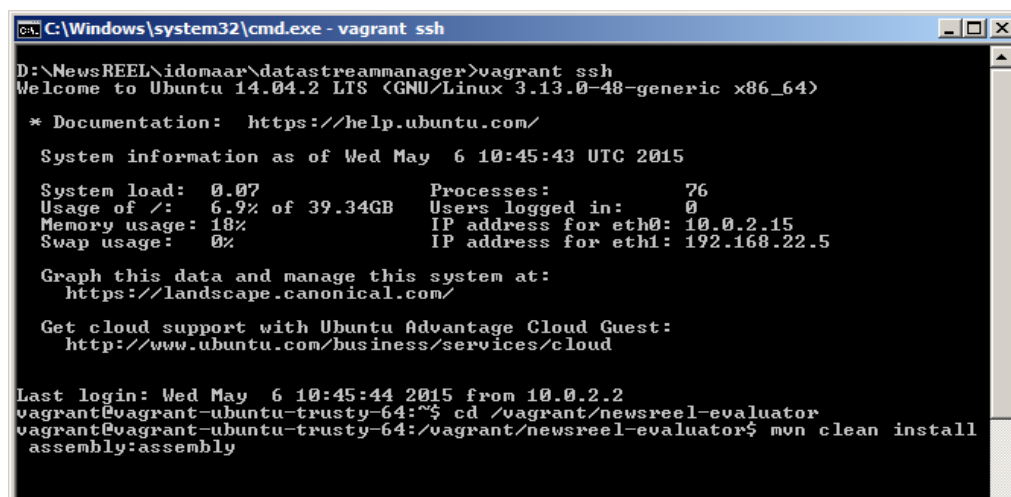
```
D:\NewsREEL\idomaar> cd datastreammanager
D:\NewsREEL\idomaar\datastreammanager> vagrant up
```

Login to the created virtual machine. For this purpose, type in the current directory the comment vagrant ssh. The tool vagrant automatically provides the correct user name and credentials.

```
[~]$ vagrant ssh
```

Build the evaluator:

```
[~]$ cd /vagrant/newsreel-evaluator
[/vagrant/newsreel-evaluator]$ mvn clean install assembly:assembly
```



```
C:\Windows\system32\cmd.exe - vagrant ssh
D:\NewsREEL\idomaar\datastreammanager>vagrant ssh
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-48-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

System information as of Wed May  6 10:45:43 UTC 2015

System load:  0.07          Processes:      76
Usage of /:   6.9% of 39.34GB    Users logged in:  0
Memory usage: 18%          IP address for eth0: 10.0.2.15
Swap usage:   0%           IP address for eth1: 192.168.22.5

Graph this data and manage this system at:
https://landscape.canonical.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

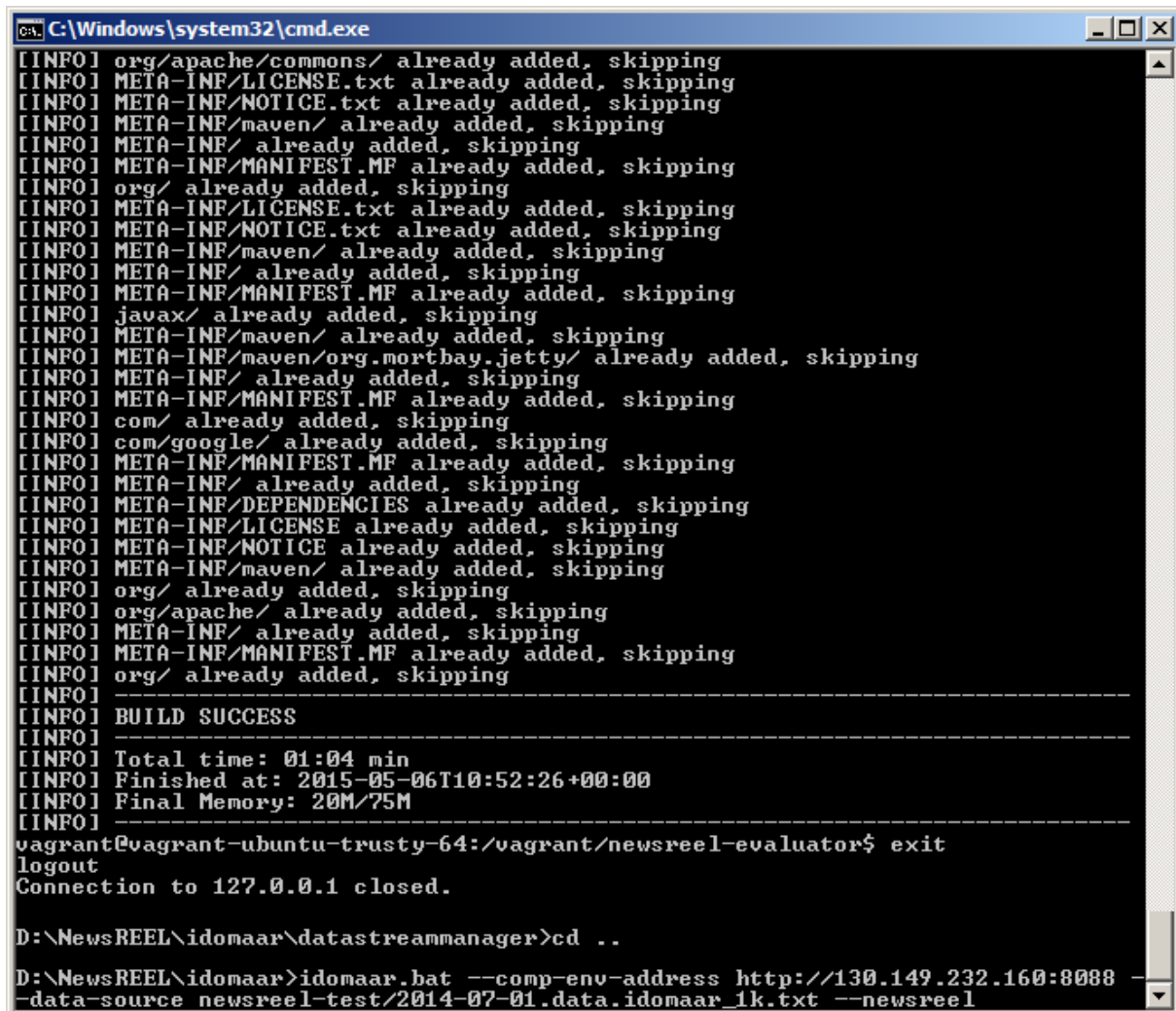
Last login: Wed May  6 10:45:44 2015 from 10.0.2.2
vagrant@vagrant-ubuntu-trusty-64:~$ cd /vagrant/newsreel-evaluator
vagrant@vagrant-ubuntu-trusty-64:/vagrant/newsreel-evaluator$ mvn clean install assembly:assembly
```

Step5: Run the Idomaar-based evaluation.

Run the Idomaar-based evaluation. Change into the `idomaar` directory and start the `idomaar` skript.

```
D:\NewsREEL\idomaar> idomaar.bat --comp-env-address
http://130.149.232.160:8088 --data-source newsreel-test/2014-07-
01.data.idomaar_1k.txt --newsreel
```

Note: The IP address used in the example (130.149.232.160) must be replaced by the IP address of the machine that run the computing environment, e.g. localhost.



```
C:\Windows\system32\cmd.exe
[INFO] org/apache/commons/ already added, skipping
[INFO] META-INF/LICENSE.txt already added, skipping
[INFO] META-INF/NOTICE.txt already added, skipping
[INFO] META-INF/maven/ already added, skipping
[INFO] META-INF/ already added, skipping
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] org/ already added, skipping
[INFO] META-INF/LICENSE.txt already added, skipping
[INFO] META-INF/NOTICE.txt already added, skipping
[INFO] META-INF/maven/ already added, skipping
[INFO] META-INF/ already added, skipping
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] javax/ already added, skipping
[INFO] META-INF/maven/ already added, skipping
[INFO] META-INF/maven/org.mortbay.jetty/ already added, skipping
[INFO] META-INF/ already added, skipping
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] com/ already added, skipping
[INFO] com/google/ already added, skipping
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] META-INF/ already added, skipping
[INFO] META-INF/DEPENDENCIES already added, skipping
[INFO] META-INF/LICENSE already added, skipping
[INFO] META-INF/NOTICE already added, skipping
[INFO] META-INF/maven/ already added, skipping
[INFO] org/ already added, skipping
[INFO] org/apache/ already added, skipping
[INFO] META-INF/ already added, skipping
[INFO] META-INF/MANIFEST.MF already added, skipping
[INFO] org/ already added, skipping
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 01:04 min
[INFO] Finished at: 2015-05-06T10:52:26+00:00
[INFO] Final Memory: 20M/75M
[INFO] -----
vagrant@vagrant-ubuntu-trusty-64:/vagrant/newsreel-evaluator$ exit
logout
Connection to 127.0.0.1 closed.

D:\NewsREEL\idomaar\datastreammanager>cd ..

D:\NewsREEL\idomaar>idomaar.bat --comp-env-address http://130.149.232.160:8088 --
-data-source newsreel-test/2014-07-01.data.idomaar_1k.txt --newsreel
```

The result of the evaluation is a table listing the CTR for different news-domains (labeled with the ID).

Note: The virtual machine and the host machine communicate with each other using port 8088. This works here since vagrant defines a forwarding rule for port 8088. In the `vagrantfile` the following rule is defined:

```
node_config.vm.network "forwarded_port", guest: 8088, host: 8088
```

```
INFO [datastream] Evaluator received END sign, stopping.
INFO [datastream]
INFO [datastream] Evaluation results
INFO [datastream] =====
INFO [datastream] 596      [2, 334, 0]      5 o/oo
INFO [datastream] 694      [0, 24, 0]      0 o/oo
INFO [datastream] 1677     [2, 448, 0]     4 o/oo
INFO [datastream] all      [4, 806, 0]     4 o/oo
INFO [datastream] mean/min/max/stdDev/n      10.362962962962962      3.0277.0
INFO [datastream] 25.33053358012503      135
INFO [orchestrator] Shutting down recommendation managers ...
INFO [orchestrator:execute] Stopping recommendation manager RMO
INFO [datastream] * Flume agent is not running
INFO [computing-environment-http] Closing ...
INFO [orchestrator] Orchestrator shutdown.
INFO [orchestrator] Finished.
Connection to 127.0.0.1 closed.
D:\NewsREEL\idomaar\datastreammanager>
```

The console shows the evaluation results for the current recommender algorithm.