PERSON Tutorial

v1.0

In this tutorial, we first explain the prerequisites. Then, we discuss the main steps for evaluation using PERSON with the code provided. Finally, we discuss the experiment-specific instructions for each experiment of the paper.

Prerequisites

In this tutorial, we assume that Lucene 7.2.1, Mallet 2.0.7 (for topic modeling) and Ubuntu OS are used and we have two input files “papers_giant.txt” (the papers file) and “authors_graph_giant.csv” (the social network file, possibly the co-authorship network) formatted as:

papers_giant.txt: The information of papers and references
paper ID Is paper merged original paper ID blank blank blank blank title abstract time (only the year part is important) blank references to papers out of the dataset references to papers inside the dataset author IDs ...
Example: 204321 0 285551   Advanced control flows for flexible graphical user interfaces: or, growing GUIs on trees or, bookmarking GUIs Web and GUI programs represent two extremely common and popular modes of human-computer interaction. Many GUI programs share the Web's notion of browsing through data- and decision-trees. This paper compares the user's

browsing power in the two cases and illustrates that many GUI programs fall short of the Web's power to clone windows and bookmark applications. It identifies a key implementation problem that GUI programs must overcome to provide this power. It then describes a theoretically well-founded programming pattern, which we have automated, that endows GUI programs with these capabilities. The paper provides concrete examples of the transformation in action. Tue Jan 01 00:00:00 IRST 2002 101286,124674,896802,877263,436439,843361,1099512,447503,532031,323630,409577,478599,1316984, 75696,203162,337930,336277,99106,321127,428853,452119,200809,411267,299602,20923,

authors_graph_giant.csv: The information of authors and their co-authors.
Src,Dst,Weight
Example: 772002,772003,1

To convert “authors_giant.txt” in the dataset to “authors_graph_giant.csv”, run the main method of AMinerPreprocessor (set inputFile and outputFile variables accordingly).

Configurations of the application are placed at ir.ac.ut.iis.person.Configs
First set datasetName, datasetRoot, and database_name to the correct values. “papers_giant.txt” and “authors_graph_giant.csv” must be placed at the root address of the dataset. Also, if needed, create a database in MySQL with the name indicated in database_name and set username, password, and address of the MySQL in ir.ac.ut.iis.person.algorithms.social_textual.MySQLConnector.

Make sure to provide enough heap space for the runs. -Xmx8000m may be reasonable choice.

It is recommended to replace the stopwords list, “src/main/resources/stoplists/en.txt”, with “https://github.com/mimno/Mallet/blob/master/stoplists/en.txt”. We used the list of Mallet v2.0.7 in our experiments. We did not include this file in the codes because of the copyright reasons.

In the following, whenever we state that “run command X in the main method of ir.ac.ut.iis.person.Main”, we mean that run the main method with the line containing X uncommented and the other lines commented.

Main Steps:

The main steps for conducting evaluation using PERSON are:

1. Indexing the dataset
2. Creating the queries file
3. Implementing the algorithms being compared

4. Performing the evaluation

Indexing

If none of the algorithms being compared needs the topic information of the documents, run the following command in the main method of ir.ac.ut.iis.person.Main to index the dataset:

```
createIndex(Configs.topicsName, Configs.profileTopicsDBTable,   Configs.RunStage.CREATE_INDEXES);
```

If any of the algorithms being compared need the topic information of the documents perform the following steps instead:

a) Extracting the topics:
   1) Use main in ir.ac.ut.iis.person.others.ExportDataset to export the documents as the input for Mallet.
   2) Set MEMORY=10g (or any other required amount) in bin/mallet
   3) Use the following command to convert the documents into Mallet format (assuming that "docs-mallet.txt" is put in a folder inside the mallet root):

      ```
      ../bin/mallet import-file --input docs-mallet.txt --output input.mallet --keep-sequence --remove-stopwords
      ```
   4) Use the following command to learn the topics:

      ```
      ../bin/mallet train-topics --input input.mallet --output-model model.mallet --output-state state.mallet --inferencer-filename inferencer.mallet --evaluator-filename evaluator.mallet --output-topic-keys topic-keys.mallet --topic-word-weights-file topic-word-weights.mallet --word-topic-counts-file word-topic-counts.mallet --xml-topic-report xml-topic-report.xml --xml-topic-phrase-report topic-phrase-report.xml --output-doc-topics doc-topics.mallet --num-topics 100 --num-threads 4 --optimize-interval 10 --optimize-burn-in 20 --use-symmetric-alpha false
      ```

b) Set topicsName in Configs. For example, if the topicsName is "100_AsymmetricAlpha" the topics files must be put under "DatasetRoot/topics/100_AsymmetricAlpha/"
c) The address should include "doc-topics.mallet" which is the output of Mallet and consists of the distribution of topics in each document and model.mallet which is the Mallet model.
d) Run the following command in the main method of ir.ac.ut.iis.person.Main:

   ```
   createIndex(Configs.topicsName, Configs.profileTopicsDBTable,
   Configs.RunStage.CREATE_INDEXES_WITH_TOPICS_STEP1);
   ```
e) Run the following command in the main method of ir.ac.ut.iis.person.Main:

   ```
   createIndex(Configs.topicsName, Configs.profileTopicsDBTable,
   Configs.RunStage.CREATE_INDEXES_WITH_TOPICS_STEP2);
   ```

Generating Queries

To create the queries run the following command in the main method of ir.ac.ut.iis.person.Main:

```
createIndex(Configs.topicsName, Configs.profileTopicsDBTable, Configs.RunStage.CREATE_QUERIES);
```

Performing Evaluation

To perform the evaluation, first set Configs.indexName according to the name of the index folder (e.g., "index_100_AsymmetricAlpha"). Then, initialize the application with

```
new PapersMain().main(name);
```

in which name is the name you specify for the run. Then add the algorithms to be compared. Some methods are provided in ir.ac.ut.iis.person.AddSearchers for adding some algorithms. Finally call the retrieve method in ir.ac.ut.iis.person.Main. Then, see the results files in the corresponding folder in the “results” folder.

Experiment-Specific Instructions

In this section, we explain the experiment-specific instructions for each experiment in the paper. Some helper methods are added in ir.ac.ut.iis.person.Main to initialize the configurations for each experiment and perform the experiment. Please note that the exact values of the results may vary depending on the Lucene version used, random seeds used, etc.

Experiment I

Run

```
        expAxioms();
```

Because of the changes in Lucene 7 (See https://issues.apache.org/jira/browse/LUCENE-7347 and also compare https://lucene.apache.org/core/6_6_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html with https://lucene.apache.org/core/7_2_0/core/org/apache/lucene/search/similarities/TFIDFSimilarity.html), the results of Experiment I may be different with the results in the paper.

Experiment III

First, put “topic-keys.mallet” and “inferencer.mallet” in the corresponding topics folder. Also use main in ir.ac.ut.iis.person.topics.InstanceClassifier to produce “alphabet.txt”. Then, run the following command in the main method of ir.ac.ut.iis.person.Main:

```
        expMoreAxioms();
```

Experiment IV

Run the following command in the main method of ir.ac.ut.iis.person.Main:

```
    expCampos();
```

Experiments V-VII

Run the following command in the main method of ir.ac.ut.iis.person.Main:

```
        expCamposTau()
```

However, since running this experiment is more complicated and requires other steps (e.g., finding the general topics and not considering the related documents as query papers), if you need running this experiment, please contact us at shayantabrizi [at] gmail [D.O.T] com.

Experiment VIII

First we need to prepare the required database tables for Social-Textual:
Run the following command in the main method of ir.ac.ut.iis.person.Main:

```
        createIndex(Configs.topicsName, Configs.profileTopicsDBTable,
        Configs.RunStage.CREATE_SOCIAL_TEXTUAL_DATABASE);
```

Remove the comma at the end of the resulting files “coauthors_giant.sql” and “papers_giant.sql” before the semicolons.

Create the required tables in MySQL:

```
        CREATE TABLE `coauthors` (
          `uid1` int(11) NOT NULL,
          `uid2` int(11) NOT NULL,
          `weight` int(11) DEFAULT NULL,
          PRIMARY KEY (`uid1`,`uid2`),
          KEY `IDX_uid1` (`uid1`),
          KEY `IDX_uid2` (`uid2`)
        ) ENGINE=InnoDB DEFAULT CHARSET=utf8;

        CREATE TABLE `UserPapers` (
          `uid` int(11) NOT NULL,
          `paperId` int(11) NOT NULL,
          `numOfAuthors` int(11) DEFAULT NULL,
          `numberOfAuthorsCoauthors` int(11) DEFAULT NULL,
          PRIMARY KEY (`uid`,`paperId`)
        ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Because of the large sizes of the sql files, you may also need to increase the allowed packet size in MySQL:

1. In “/etc/mysql/mysql.conf.d/mysqld.cnf” (may be at a different place), set

```
        max_allowed_packet      = 128M
```

2. restart MySQL:

```
        sudo service mysql restart
```

The run the SQL files generated (Assuming the user is root and the database name is “aminerDB”):

```
        mysql -u root -p aminerDB <  papers_giant.sql
        mysql -u root -p aminerDB <  coauthors_giant.sql
```

Now, use these queries to enrich table UserPapers:

```
create table temp(uid1 int(11), cnt int(11), index(uid1)) select uid1, count(*) as cnt from coauthors group by uid1;
update UserPapers set NumberOfAuthorsCoauthors = (select cnt from temp where uid1=uid);
```

Then, run the following command in the main method of ir.ac.ut.iis.person.Main:

```
        expCompare();
```

Experiment X Part III

Run the following command in the main method of ir.ac.ut.iis.person.Main:

```
    expRobustness();
```

Appendix A

There are some classes that may be useful:

1. ir.ac.ut.iis.person.others.ResultConverter: Converting results files to other formats.
2. ir.ac.ut.iis.person.EvaluatorComparator: For comparing the results of ASPIRE and those of PERSON.