

Description of Software and Data Sets for Simulating Tagging Models

Klaas Dellschaft

klaasd@uni-koblenz.de

1 Introduction

This file gives an introduction to the the software and the data sets which have been used in [1]. In [1], more details about the simulation models and the data sets is available. The software, its source code as well as the data sets are contained in the jar-file `websci2010-taggingmodels.jar` which can be downloaded from `http://west.uni-koblenz.de/Research/DataSets/websci2010`. Source code and data sets can be extracted from the jar file with the help of any zip utility.

2 Software

2.1 Simulation – GUI

The GUI for repeating the simulations done in [1] can be started with the following call:

```
java -Xmx512m -jar websci2010-taggingmodels.jar
```

It can be used for simulating the Epistemic Model as well as the Yule-Simon Model with Memory (see Fig. 1). For reproducing the results from [1], do the following steps:

1. Select the model and the stream for which you want to reproduce the results (*Stream to be simulated*).
2. Select the simulation of whole postings (*Simulation of postings*).
3. Enter how often the simulations should be repeated (*Repetitions of the simulation*).
4. Choose an output directory where the simulated streams should be saved. In this directory, you will find afterwards files containing the raw streams. The convention for the automatically generated file names is *epistemic_IValue_hValue_nValue_runNr.stream* for the Epistemic Model or *ysm_pValue_tauValue_n0Value_runNr.stream* for the Yule-Simon Model with Memory.
5. Set the parameters of the respective model

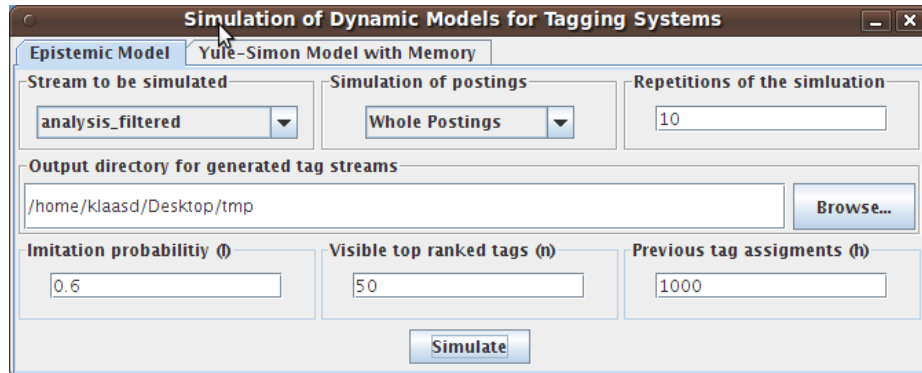


Fig. 1. Screenshot of the simulation GUI.

2.2 Simulation – Command Line

The simulations can also be started on the command line. This option gives more flexibility with regard to the simulation, e. g. other distributions of postings sizes may be used than in the GUI and streams with arbitrary length or also other streams than in [1] can be used. The command line can be started with the following call:

```
java -Xmx512m -cp websci2010-taggingmodels.jar
    de.unikold.isweb.TagStreamSimulator
```

More details about the available options are given on the command line if no further parameter is given to the above call.

2.3 Generating Plots

The two applications from above for doing the actual simulations only save the raw simulated stream. If one wants to extract the tag frequency distribution or the vocabulary growth for one or more of the streams, one has to use another application. It is started with the following call:

```
java -Xmx512m -cp websci2010-taggingmodels.jar
    de.unikold.isweb.TagStreamSaver
```

It opens a file chooser where one or more streams can be selected (see Fig. 2). The extracted plots of the tag frequency distribution as well as the vocabulary growth will be saved in the same directory as the file which contains the stream. The saved files will be named **.edf.freq*, **.fr.freq* and **.growth*. More details about the used file format is available in Section 4.

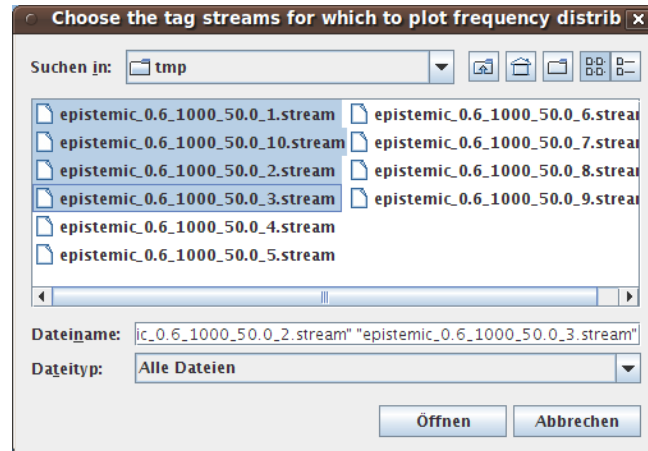


Fig. 2. Screenshot of the file dialog for extracting the frequency distribution and vocabulary growth from stream files.

2.4 Applying the Smirnov Test

For comparing the tag frequency distribution of a simulated stream with that of the original stream, one can use the following application:

```
java -Xmx512m -cp websci2010-taggingmodels.jar
    de.unikold.isweb.StreamComparison
```

If it is called without any further parameters, a file chooser dialog appears where one can first select the file which contains the original stream. Then, a second file chooser appears where one or more simulated streams can be selected which should be compared with the original stream.

There is also the option to use the application as a pure command line tool. In that case, the first parameter passed to the application is interpreted as the file name of the original stream. All further parameters are then interpreted as the file names of the simulated streams.

2.5 Source Code

The source code of all classes needed for the applications from above is contained in the jar file itself. After extracting the content of the jar file with the help of any zip utility, the relevant source code is located in `de/*`. The required libraries for compiling the source code are contained in `lib/*`.

3 Data Sets

The jar file *websci2010-taggingmodels.jar* also contains the filtered and unfiltered streams which have been used in [1]. After extracting the content of the jar file with the help of

any zip utility, one has to also extract the content of the file *main/main.jar*. Alternatively, one may directly download and extract the file *datasets.zip* if one is only interested in the datasets and not in the software described above.

The directories in the extracted *main/main.jar* or *datasets.zip* are organized as follows:

- *delicious/** This directory contains all 10 stream pairs which have been extracted from Delicious.
- *bibsonomy/** This directory contains all 5 stream pairs which have been extracted from Bibsonomy.
- *webcorpora/** This directory contains the files with the occurrence probabilities in the 15 web corpora which have been crawled for simulating the background knowledge in the Epistemic Model.
- *average.posting* Contains the probabilities of observing postings with a certain size in the overall Delicious data set. This file is used for simulating the postings in the Epistemic Model and the Yule-Simon Model with Memory.

4 Used File Formats

This section describes the format of the files used for storing the raw stream information or the different kind of plots for the tag frequency distribution or the vocabulary growth. All files are saved as plain text files and can be read with any text editor.

4.1 *.stream

The **.stream* files are used for saving the raw stream of tag assignments. It is a simple text file in which each line corresponds to a single tag assignment. Each line consists of four columns which are separated by a tab character:

1. **Posting ID** The first column contains the integer ID of the posting, to which this tag assignment belongs.
2. **Tag ID** The second column contains the integer ID of the tag used in this tag assignment.
3. **User ID** The third column contains the integer ID of the user who assigned the tag.
4. **Resource ID** The fourth column contains the integer ID of the resource to which the tag has been assigned.

In the simulated streams, the user ID and resource ID are not simulated in a meaningful way. Thus, the simulated streams will always contain a lower number of resources and users than the original stream. Classes for reading such a stream are available in the *websci2010-taggingmodels.jar*. The following line of code can be used for opening such a file in Java:

```
PostingStream stream = (PostingStream) TASStream.loadFromURL(url);
```

4.2 *.edf.freq

The *.*edf.freq* files are used for saving the tag frequency distribution of a stream in its empirical distribution function form. It is a simple text file in which each line corresponds to a x- and y-value in the step function of the empirical distribution function. The first column in the file contains the x-value and the second column the corresponding y-value. The values are ordered by increasing x-value. Only those data points are given, where the y-value of the step function changes.

4.3 *.fr.freq

The *.*fr.freq* files are used for saving the tag frequency distribution of a stream in its frequency-rank plot form. It is a simple text file in which each line corresponds to a x- and y-value in the frequency-rank plot. The first column in the file contains the x-value and the second column the corresponding y-value. The values are ordered by increasing x-value.

4.4 *.growth

The *.*growth* files are used for saving the vocabulary growth in a stream. It is a simple text file in which each line corresponds to a x- and y-value in the plot of the vocabulary growth. The first column in the file contains the x-value and the second column the y-value. The values are ordered by increasing x-value.

4.5 *.occ

The *.*occ* files are used for saving the occurrence probabilities in a corpus of web documents. It is a simple text file in which each line corresponds to the occurrence probability of one specific word. The columns within a line are separated by the tab character. The first column contains the tag ID which will be used for this word in the simulated tag streams (i. e. it links the web corpus to the second column in the *.*stream* files). The second column contains the word itself. The third column gives the occurrence probability of the word in the corresponding web corpus. The entries are ordered by decreasing occurrence probability of the words.

4.6 *.posting

The *.*posting* file is used for saving the probabilities of observing a posting of a certain size. It is a simple text file in which each line corresponds to the probability of observing a specific posting size. The columns within a line are separated by the tab character. The first column contains the size of a posting and the second column the probability of observing a posting of that size. The entries are ordered by increasing posting size.

References

1. Dellschaft, K., Staab, S.: On differences in the tagging behavior of spammers and regular users. In: Proceedings of the Web Science Conference 2010. (2010) <http://www.isweb.uni-koblenz.de/files/publications/Dellschaft2010ODI.pdf>.