Content-Based Tag Recommender
Second Term 2014

Home Assignment

Due Sunday, October 5th of 2014

Assignment leaders:
Christoph Trattner (Know-Center, Austria)
Dominik Kowald (Know-Center, Austria)
Hans Findel (PUC, Chile)
Denis Parra (PUC, Chile)

Introduction

The aim of this home assignment is to get used with a tag recommender framework (`https://github.com/learning-layers/TagRec`) written in Java and to implement and evaluate a content-based tag recommender on top of it using a dataset sample of the social bookmark and publication sharing system BibSonomy (`http://www.bibsonomy.org/`).

Provided files

The file `https://github.com/learning-layers/TagRec/archive/v1.1.2.zip` contains the source code and the dataset (`data/csv/bib_core`) for this assignment. Moreover, this archive also contains an eclipse project file and thus, can be easily imported into Eclipse Luna (`https://www.eclipse.org/`). Please also read the documentation at `https://github.com/learning-layers/TagRec` and `http://www.christophtrattner.info/pubs/ht241-kowald.pdf`.

Task 1: Read in the data

The first task is to extend the `common/Bookmark` and `file/BookmarkReader` classes in order to make them capable of content-based data. The content-based data is located in the last two columns of the input files (`title;description`). Please note that both classes contain `TODO` placeholders where you should insert your code.

Task 2: Implement a content-based tag recommender

The next task is to implement a content-based tag recommender based on a method/algorithm you choose (see *References*). This has to be done in the `processing/ContentBasedCalculator` skeleton class in the method `getRankedTagList` (see also the `TODO` placeholders here). Of course you are allowed to add as many helper methods and classes as you want. Please also have a look at the utility functions in `common/Utilities`.

In order to improve your content-based tag recommender, you can also mix it with (parts of) other already implemented folksonomy-based approaches (we suggest here `processing/LanguageModelCalculator` or `processing/ActCalculator`).

Metric	Value
Recall	0.712
Precision	0.229
F1-score	0.347
MRR	0.408
MAP	0.523
nDCG	0.614
User-Coverage	1.0

Table 1: Results (@10) for the time-based recommender approach BLL+C (= your baseline).

Task 3: Evaluate your tag recommender

In order to test and evaluate your tag recommender approach just execute the `test/Pipeline` main class. It already contains the call to your recommender (`startContentBasedCalculator`) as well as the calls to all other recommenders (just uncomment the code lines to test them). As described on `https://github.com/learning-layers/TagRec`, the result of the evaluation is a file created in `data/metrics/bib_core` that contains the accuracy of your recommender in terms of different information retrieval metrics.

The metrics to beat can be found in Table 1 (last line of the created file). The results have been created using the time-dependent BLL+C algorithm [Kowald, 2014b].

Task 4: Document and explain your tag recommender strategy and results

You also have to create a `report.pdf` file where you document and explain your content-based tag recommender strategy in detail. This document should also contain the results of your evaluation.

Submission

Please provide your assignment in form of a `.zip` archive with the filename `surname_recommender.zip` (e.g., `trattner_recommender.zip`). This archive should contain all the code files you have edited/created (thus, at least `common/Bookmark`, `file/BookmarkReader` and `processing/ContentBasedCalculator`) as well as your report file `report.pdf`. Please also note that your code must be executable with Java SE Development Kit 8 (`http://www.oracle.com/technetwork/java/index.html`)on Eclipse Luna (`https://www.eclipse.org/`).

Please send your submission per e-mail to Christoph Trattner (`trattner.christoph@gmail.com`), Dominik Kowald (`dkowald@know-center.at`), Hans Findel (`hanstrax@gmail.com`) and Denis Parra (`denisparra@gmail.com`). **Important:** your e-mail subject must include `[CB_recommender]`.

References

- Recommender systems Wikipedia article: http://en.wikipedia.org/wiki/Recommender_system
- D. Kowald, E. Lacic, and C. Trattner. Tagrec: Towards a standardized tag recommender benchmarking framework. In Proceedings of the 25th ACM Conference on Hypertext and Social Media, HT'14, New York, NY, USA, 2014a. ACM.: http://www.christophtrattner.info/pubs/ht241-kowald.pdf
- D. Kowald, P. Seitlinger, C. Trattner, and T. Ley. Long Time No See: The Probability of Reusing Tags as a Function of Frequency and Recency. In Proceedings of the 23rd international conference on World Wide Web Companion, WWW '14, Seoul, Korea, 2014b. ACM.: http://www2014.kr/wp-content/uploads/2014/05/companion_p463.pdf
- M. Lipczak: Hybrid Tag Recommendation in Collaborative Tagging Systems. Dissertation at Dalhousie University Halifax, Nova Scotia, 2012: http://dalspace.library.dal.ca/bitstream/handle/10222/14735/Lipczak,Marek,PhD,CS,March2012.pdf.