Universität Hamburg
Department Informatik
Wissenschaftliches Rechnen
Dr. J. Kunkel, H. Hatzel

Exercise Sheet 9
Big Data Analytics
WiSe 2016/2017
Due Date: 2016-12-23, 450 P

Please document your code, without sufficent documentation you won't recive any points.

1 HBase Wikipedia Data Model & Import (Python) (180 P)

Goal of this task is to define a data model to

1. store the absolute word frequencies in HBase.
2. import a few articles of our data file `/home/bigdata/9/enwiki-clean-10MiB.csv` in HBase.
3. create a Python command line program which identifies all articles that contain a certain word.
4. discuss how the schema could be modified to contain a reverse index for the positions of a word within an article, e.g., the word HBase occurs on position 0 and 38 in the Text "HBase is better columnar storage than HBase.".

When you define the data model in 1) and 4), consider alternative mappings to store the information. What would be the advantage and disadvantage of adding an index and/or reverse index?

1.1 Hints

Please use your name for the table to prevent collisions with other groups.
We will use HappyBase `https://happybase.readthedocs.io/en/latest/` to access HBase from Python.
The Thrift server is started on Abu1 on Port 9090.

Submission:

`1-hbase-model.pdf`	The data model for Task 1) and the extended model for Task 4).
`1-hbase-import.py`	The HBase import program that also creates the data model.
`1-hbase-query.py`	The script to find all articles that contain the word.

2 Classification of new Wikipedia articles (R or Python) (210 P)

Imagine we try to propose categories for a Wikipedia article that lacks this information. Develop an algorithm and evaluate its performance on existing data.
We could use supervised learning to achieve this goal as follows:

- Load the CSV file `/home/bigdata/9/enwiki-categories.csv` containing article name and categories.
- Split the articles (or a subset of them) into training, validation and test set.
- Train a k-nearest neighbor classifier with the training data and one other model of your choice. Features for an article could be the relative word frequency, i.e., two articles are similar if all individual word frequencies are similar. You can also user other metrics of your choice.
- Check the accuracy of the predictions on the validation sets and tune your models.
- Assess the expected accuracy of your tuned models on the test set.

- Inspect a few (random) articles and compare the suggestions of your classifier with the available data.

You may use Python or R to solve this task. Create a lab notebook with your code, results and interpretation.

Submission:

`2-wikipedia-knn.pdf` A lab notebook with your code, analysis and results.

3 Data Model for Document Storage (Theory) (60 P)

In this task, create a data model to store raw Wikipedia articles in MongoDB together with the derived data such as word-frequency, extracted categories, etc. per article. Model the derived data in such a way that new derived operations / data cleaning can easily extend the existing data.

Submission:

`3-document-model.pdf` Your documented object model.