Programming Assignment 2

Document Summarization using TF/IDF Scores

Due: March 26, 2018 By 12:00PM
Submission: via Canvas, individual submission

Objectives

The goal of this programming assignment is to enable you to gain experience in:

- Creating an authorship identification system based on the similarity of the author's attributes
- Calculating TF/IDF scores using MapReduce
- Document summarization using MapReduce

1 Introduction

The goal of this assignment is to take a given document, and extract sentences that best summarize the document. The system should use word uni-grams that was used in programming assignment 1. We will calculate the level of importance for each sentence using TF-IDF (Term Frequency Inverse Document Frequency).

1.1 Term Frequency

Suppose we have a collection of documents written by M authors. This collection of documents may contain multiple Wikipedia articles. For a Wikipedia article j, we define f_{ij} to be the frequency (Number of occurrences) of term (word) i in the Wikipedia article j.

$$TF_{ij}=0.5+0.5(f_{ij}/max_k f_{kj})$$

In this assignment, we use the augmented TF to prevent a bias towards longer documents. E.g. raw frequency divided by the maximum raw frequency of any term k in the article j. During this process, you should not eliminate stop words. The most frequent term in the sub-collection will have an augmented TF value of 1.

1.2 Inverted Document Frequency

Suppose that term i appears in n_i articles within the corpus. For this assignment, we define the IDF_I, as:

$$IDF_i = log_{10}(N/n_i)$$

where, N is the total number of articles.

1.3 TF.IDF value

The TF-IDF score is defined as TF_{ij} x IDF_i. The terms with the highest TF-IDF score are considered the best words that characterize the document.

1.4 Scoring Each Sentence

Using the result of performing calculations outlined in sections 1.1 through 1.3, we score each sentence in the article. For each sentence S_k, calculate *Sentence.TF.IDF*,

$$Sentence.TF.IDF(S_k) = \sum top\ n\ TF.IDF(word_i)$$

To avoid the case that any long sentence gets the highest total score, you should add only the n highest TF.IDF values within a sentence to get Sentence.TF.IDF. In this assignment, use n=5 as the maximum number of TF.IDF values that you will add. If the number of words included in a sentence is smaller than 5, you must add all of the TF.IDF values available for that sentence. If there is(are) repeated words, count only once.

1.5 Generating a Summary

Suppose that there are 50 sentences ($S_0 \sim S_{49}$). Based on the *Sentence.TF.IDF* calculation described in section 1.4, you have 50 values:,

$Sentence.TF.IDF(S_0)$
$Sentence.TF.IDF(S_1)$
$Sentence.TF.IDF(S_2)$
…
$Sentence.TF.IDF(S_{49})$

To generate a summary, select the top 3 sentences and list them based on the original sequence of those sentences within the document.

2. Software Requirements

Your software should perform multiple steps;

(1) You should calculate the *TF*, *IDF*, and *TF-IDF* values for all terms for all sub-collections in your corpus. **You are required to use MapReduce(s) for this step**. Custom implementations without using MapReduce is disallowed.
(2) Create the summaries of articles (Use 1G data files).
(3) You should store the results in a HDFS file.
(4) For a given article (GTA will provide an article for the demo), your software should be able to generate a summary using values generated in (1). You do not need to re-calculate IDF for this step. **You are required to use MapReduce for this step**. Again, custom implementations that do not use MapReduce is disallowed.

3 Input data

The format of input data is identical to the one in PA1. Input data file is available at this link.
You are not required to submit the output of your software on running on this 1Gb file. Keep your intermediate outputs stored in your HDFS for use later on in Software Requirements (4).

You will also find a smaller data file for testing out your software on this link. This file would be about several MBs in size and is to be used for testing purposes only. This is for testing purposes only.

4. Submission

This assignment must be done **individually**. Please submit the tar ball of your source files along with any jar you need to run your software via Canvas. The source files should include your java code for Mapper/Reducer functions and any script file that you will use for the demo. You will download your jars from Canvas for the demo. Do not miss any files. During your demo, you are not allowed to use any file outside your Canvas submission.

5. Grading

Each of the submissions will be graded based on the demonstration of your software and interview. During the demonstration, you should present:

Step 1. TF/IDF calculation [**4 points**]
Step 2. Creating summaries of articles in the 1G dataset. [4 points]
Step 3. Creating a summary of an arbitrary article given by GTA [2 points]

Demos include short interviews about your software design and implementation details. Each question and answer will count toward your score. **This assignment will account for 10% of your final course grade.**

6. Late policy

Please check the late policy posted on the course web page.