Uppsala University

Assignment 2
Report

Author:
Jinglin Gao

February 10, 2023

1 Introduction

This report contains the answers for each question in each task as well as plots with explanation to explain the results I got.

2 Task 1.1.Word count example in local mode

2.a

There are two files in output. One is '_SUCCESS' and the other one is 'part-r-00000'.
'_SUCCESS' is a flag file that indicates the job has completed successfully. This file is an empty file with a name starting with an underscore, and its presence is used to indicate that the job has completed successfully.
'part-r-00000' is the actual output file containing the word count results. The number of times each word (including numbers) appears in the text is counted. And the count is not only case-sensitive but also distinguishing pattern.

2.b

The word 'Discovery' appears 5 times.

2.c

Standalone mode is usually the fastest Hadoop modes as it uses the local file system for all the input and output. In this mode, Hadoop runs on a single machine, and all the daemons (NameNode, DataNode, ResourceManager, NodeManager, etc.) run on the same machine.
The Pseudo-distributed mode is also known as a single-node cluster where both NameNode and DataNode will reside on the same machine. In this mode, all the daemons run on the same machine, but they communicate with each other as if they were running on separate nodes in a cluster.

3 Task 1.2.Setup pseudo-distributed mode

3.a

'core-site.xml' file informs Hadoop daemon where NameNode runs in the cluster. It contains the configuration settings for Hadoop Core such as I/O settings that are common to HDFS and MapReduce.

'hdfs-site.xml' file contains the configuration settings for HDFS daemons: the NameNode, the Secondary NameNode, and the DataNodes. We can configure hdfs-site.xml to specify default block replication and permission checking on HDFS. The actual number of replications can also be specified when the file is created.
In summary, 'core-site.xml' contains settings that are common to both HDFS and MapReduce, while 'hdfs-site.xml' contains settings specific to HDFS.

3.b

NameNode: NameNode works on the Master System. The primary purpose of NameNode is to manage all the MetaData. It stores the information of DataNode such as their Block's id and Number of Blocks.
DataNode: DataNode is a program that runs on the slave system that serves the read/write request from the client. The Data is stored in DataNode.
Jps: Jps is tool provided by Java which can be used to look at the specific processes of Hadoop.

4 Task 1.3.Word count in pseudo-distributed mode

4.a

There are two different classes in the file WordCount.java, one is TokenizerMapper and other one is IntSumReducer.
Mapper class implements the 'map' function of the MapReduce framework. The 'map' function takes input key-value pairs and produces intermediate key-value pairs.
Reduce class implements the 'reduce' function takes the intermediate key-value pairs produced by the 'map' function and aggregates the values associated with each key.
They insure that on each node the computation can be done and in final they can be reduce together.

4.b

HDFS is a distributed file system designed to store large amounts of data across multiple nodes in a Hadoop cluster. It provides a high-level of fault tolerance and scalability, allowing to store and process large amounts of data efficiently. It works like one file will be divided into several equal size pieces and store across cluster and while retrieval one daemon process will

be picking all file pieces address and serving retrival request.
Local filesystem is just one file which is stored in one location and will be having size limits and fault tolerance.

5 Task 1.4.Modified word count example

5.a

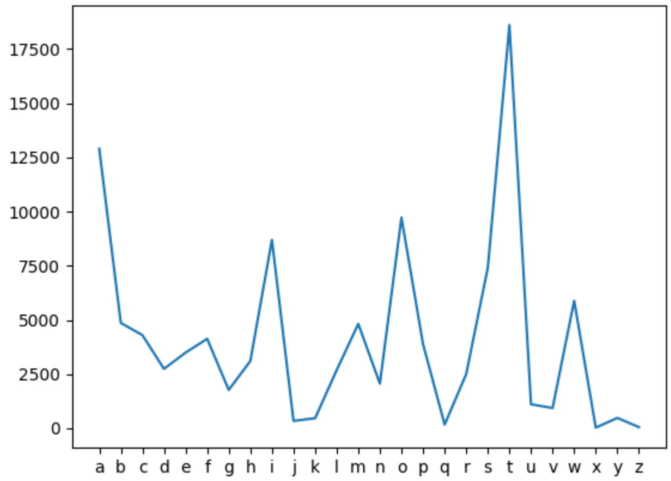

Figure 1: Counts for each letter

6 Task 1.5.NoSQL and MongoDB

6.a

For my perspective, twitter tweets is semi-structured data. Based on the documentation, the JSON is a mix of 'root-level' attributes and child objects, 'root-level' attributes are structured data however the child objects like tweet text are unstructured data.

6.b

SQL databases are relational databases, are based on the relational model and have a well-defined schema that is used to organize data into tables with rows and columns. Some of the pros of SQL databases include: Relational model, Structured data, ACID transactions. Some of the cons of SQL databases include: Scalability, Fixed schema.

NoSQL are designed to handle unstructured or semi-structured data, they are designed to scale horizontally and are often used for big data and real-time applications. Some of the pros of NoSQL databases include: Scalability, Flexible schema, Document-oriented. Some of the cons of NoSQL databases include: Unstructured data, No ACID transactions.

In conclusion, the choice between SQL and NoSQL databases depends on the specific requirements of the application. If the application requires a well-defined schema and strong consistency guarantees, an SQL database may be the best choice. On the other hand, if the application requires the ability to handle large amounts of unstructured or semi-structured data, a NoSQL database may be the better choice.

Examples for SQL databases:

Financial information, where data must be kept secure and there are strict regulations for how it is stored and managed.

Customer information, where data must be kept organized and accessible for reporting and analysis.

Examples for NoSQL databases:

Social media data, where users generate vast amounts of unstructured data, including text, images, and videos.

Log data, where there is a high volume of data being generated and the structure may vary over time.

Geospatial data, where data must be stored and queried based on geographic location.

7 Task 2.1.Analyzing twitter data using Hadoop streaming and Python

7.a

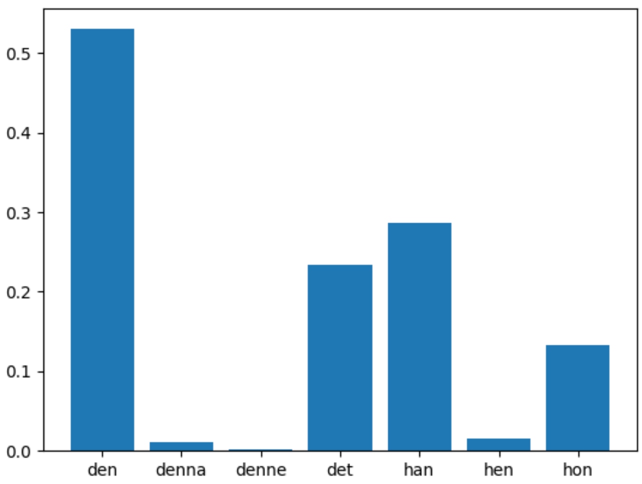

Figure 2: Counts for each pronoun

The result got from the analysis:
den 1241982
denna 24638
denne 4031
det 547225
han 671048
hen 35355
hon 310294
unique_tweets 2341577

8 Task 2.2.NoSQL and MongoDB

8.a

Use MongoDB we can directly use search queries to get the result. But the result I got from MongoDB is quiet different from the result I got before.

I use 'mongoimport' command to import the .txt file into the MongoDB database. At first I try to delete all the retweeted tweets but since there is not a lot of time I cannot finish the process. So I use a combination search query to search the pronouns. For now I have a suppose about the difference: MongoDB may count the characters inside another word as a result.

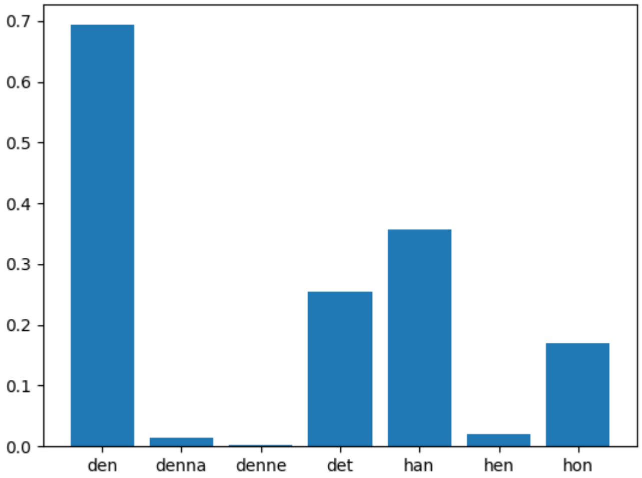

Figure 3: Counts for each letter

The result got from the analysis:
denne 6665
hon 396532
hen 46321
han 834277
det 594561
denna 31683
den 1623354
unique$_tweets$2341577