# CHAPTER 5

# ALGORITHMS

## Algorithm 1: Porter Stemmer Algorithm

**Input**

- Let T be the set of downloaded tweets.

**Output**

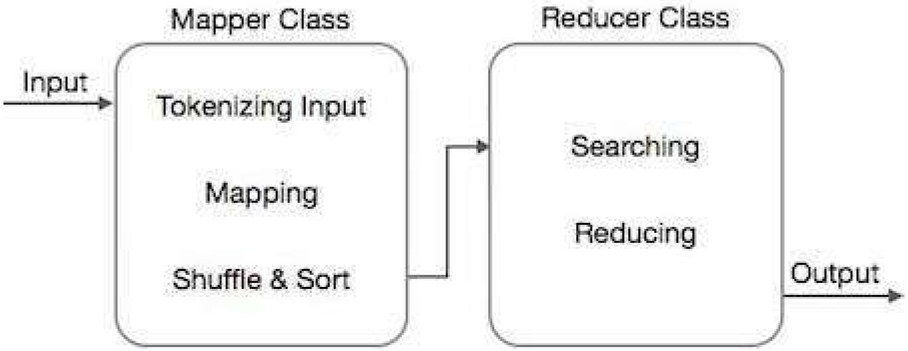
- Processed tweets with all unwanted word, space and special character removal.

## Algorithm 2: MapReduce - Algorithm

The MapReduce algorithm contains two important tasks, namely Map and Reduce.

- The map task is done by means of Mapper Class
- The reduce task is done by means of Reducer Class.

Mapper class takes the input, tokenizes it, maps and sorts it. The output of Mapper class is used as input by Reducer class, which in turn searches matching pairs and reduces them.



MapReduce implements various mathematical algorithms to divide a task into small parts and assign them to multiple systems. In technical terms, MapReduce algorithm helps in sending the Map & Reduce tasks to appropriate servers in a cluster.

These mathematical algorithms may include the following −

- Sorting
- Searching
- Indexing
- TF-IDF

## Phases of MapReduce

**Map Phase:** Input Split is not block it is a java class behind the scene with pointer to start and end location within a block.

**Mapper:** Mapper can be return in many language like Java. Mapper is a java program which is invoked by Hadoop framework once per every input split. Number of Mapper is equal to input split. Output of a mapper will be key value pair.

**Reducer:** Output of individual Mapper will be grouped by the keys and send to reducer. Number of reducer can be set by user. Each key is assigned to partition of reducer by Partition class. This Partition happen in Mapper Phase. Reducer method will call once for each key.

## Algorithm 3: Naive bays Classifier

Political orientation of users towards party, topics can be analyzed from tweets. Map Reduce version of nave bayes algorithm will be implemented to classify tweets into positive, negative and neutral classes.

Steps

1: Create a data for the classifier

1.1: Create a list of positive tweets

1.2: Create a list of negative tweets

1.3: convert this two list in to single list with two parts word array for each tweet and its type

2: Design a Classifier

2.1: Extract the word feature list from the list with its frequency count

2.2: Using this words list create feature extractor which contains the words which will matched with a dictionary created by us indicating what words are contained in the input passed

3: Training the Classifier using training dataset

3.1: Generate Table which contains positive and Negative words.

3.2: Generate Feature_Table List which contains the featured words.

4: Calculate the average for the positive and Negative Label.

5: Compare this average to identify the tweet category as positive, negative or neutral.