Thanet Sirichanyaphong z5228028

Assignment 1

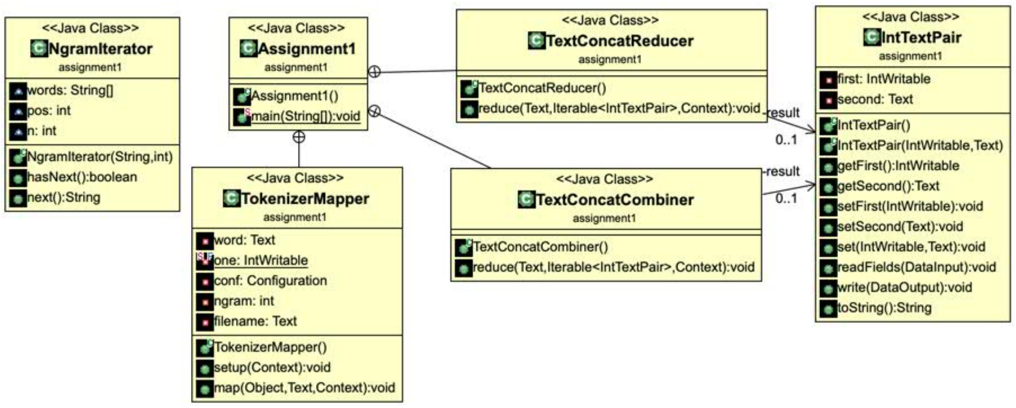

Figure1: Class Diagram

Additional Classes

1. *NgramIterator*: a java class for creating ngram-iterator from a string
2. *IntTextPair*: output value class implemented writable, this class contain pair of an integer and a string. The integer represents number of ngram-word and the string represents filenames of this word is found.

Mapper Class

1. Setup method: to get ngram parameter from configuration and filename from filesplit class.

```
public void setup(Context context) throws IOException,InterruptedException{
        conf = context.getConfiguration();
        ngram = Integer.parseInt(conf.get("ngram"));
        FileSplit fileSplit = (FileSplit) context.getInputSplit();
        filename.set(fileSplit.getPath().getName());
}
```

2. Map method: creae a NgramIterator class to split ngram-words from document. Then, iterate through each token and form a key value pair. Key is a ngram-word and value is a IntTextPair object which is a pair of Intwritable(1) and it's filename.

```
public void map(Object key, Text value, Context context) throws IOException,InterruptedException {
        NgramIterator itr = new NgramIterator(value.toString(), ngram);
        while (itr.hasNext()) {
                word.set(itr.next());
                context.write(word, new IntTextPair(one, filename));
        }
}
```

Combiner and Reducer Class

These two class is almost the same but the reducer class has a filter of minimum count for each ngram-word.

1. Only reducer class read *min_count* parameter from configuration.

```
int min_count = Integer.parseInt(conf.get("min_count"));
```

2. Create a set of string to contain list of filenames. Then, iterate through all values with respect to a key, sum up all of value from integer in a value object and add filename from string in value object to set of string.

```
Set<String> file_Set = new HashSet<String>();
for (IntTextPair val : values) {
        sum += val.getFirst().get();
        StringTokenizer itr = new StringTokenizer(val.getSecond().toString());
        while (itr.hasMoreTokens()) {
                file_Set.add(itr.nextToken());
        }
}
```

3. Only reducer class will terminate itself if the sum is lower than min_count.

```
if (sum < min_count)return;
```

4. Sort the set of string by parsing them to a treeset object and combine them into a string.

```
Set<String> sorted_file_Set = new TreeSet<String>(file_Set);
Iterator<String> file_name_itr = sorted_file_Set.iterator();
StringBuilder file_list_sb = new StringBuilder();
while (file_name_itr.hasNext()) {
        file_list_sb.append((file_list_sb.length() == 0 ? "" : " ") + file_name_itr.next());
}
```

5. Form the final key/value pairs result for each word using context.

```
result.set(new IntWritable(sum), new Text(file_list_sb.toString()));
context.write(key, result);
```