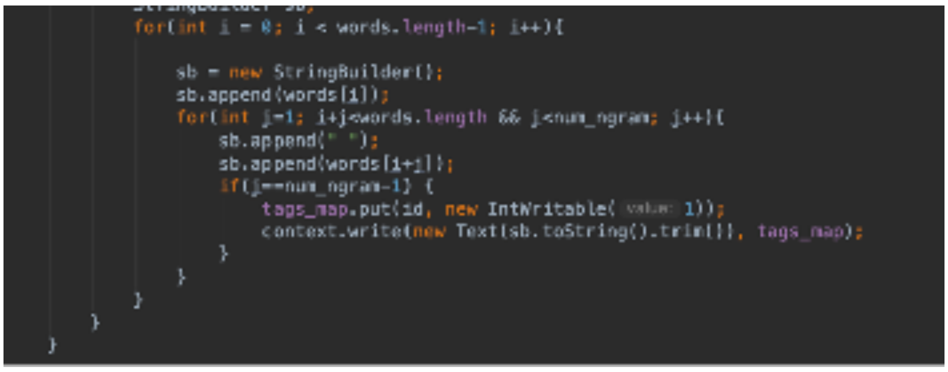
# COMP9313

Assignment1-report

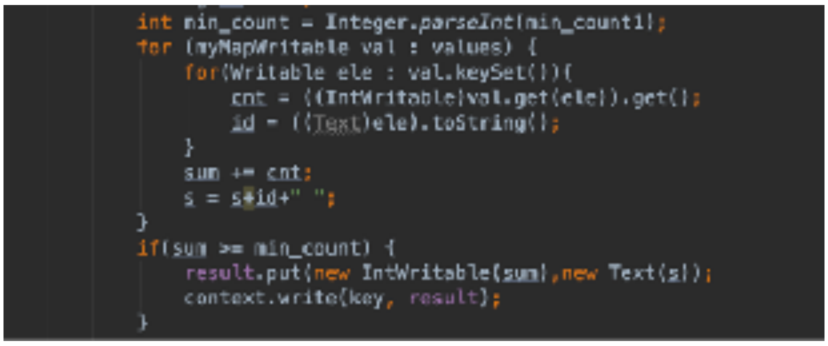
Name: Luyao Zhang
ZID: z5151973

Assignment 1 is an implementation of apache hadoop which aims to get the number of ngrams in different files. I use the apache hadoop Wordcount example and modified some of the codes to get right output.
First, we should use mapper to generate the key-value pairs. This part is similar as Wordcount codes. However, we not only get single word as a key, also need n number of words as a key, that makes a ngrams. So I add another loop in the main loop.
And for the output part, we need print the file paths which contains the key words. So I use mapwritable to save the value and the filenames. You can see the tags_map is the mapwritable. After the step, we can get the value and filenames in reducer and save in output.



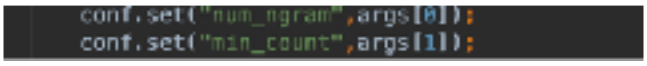
```
        for(int i = 0; i < words.length-1; i++){

            sb = new StringBuilder();
            sb.append(words[i]);
            for(int j=1; i+j<words.length && j<num_ngram; j++){
                sb.append(" ");
                sb.append(words[i+j]);
                if(j==num_ngram-1) {
                    tags_map.put(id, new IntWritable( value: 1));
                    context.write(new Text(sb.toString().trim()), tags_map);
                }
            }
        }
    }
}
```

Then the reducer part. It the same as Wordcount example. Just sum the value and get the filenames in mapwritable. Then, save the sum value and filenames in a new mapwritable, and write in the context.



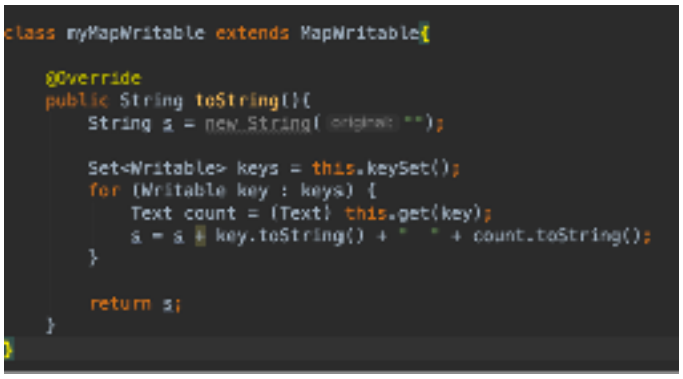
```
        int min_count = Integer.parseInt(min_count1);
        for (myMapWritable val : values) {
            for(Writable ele : val.keySet()){
                cnt = ((IntWritable)val.get(ele)).get();
                id = ((Text)ele).toString();
            }
            sum += cnt;
            s = s+id+" ";
        }
        if(sum >= min_count) {
            result.put(new IntWritable(sum),new Text(s));
            context.write(key, result);
        }
```

The main function is the same as example. However, there are two problems when I run the code. The first one is how to pass the first 2 arguments. I search in google, and I use configuration method set variable.



```
    conf.set("num_ngram",args[0]);
    conf.set("min_count",args[1]);
```

The second problem is how to print right mapwritable value. Cause the apache Hadoop's method toString cannot output the context in mapwritable. So I need to modified this method like this:



```
class myMapWritable extends MapWritable{

    @Override
    public String toString(){
        String s = new String( original: "");

        Set<Writable> keys = this.keySet();
        for (Writable key : keys) {
            Text count = (Text) this.get(key);
            s = s + key.toString() + "  " + count.toString();
        }

        return s;
    }
}
```