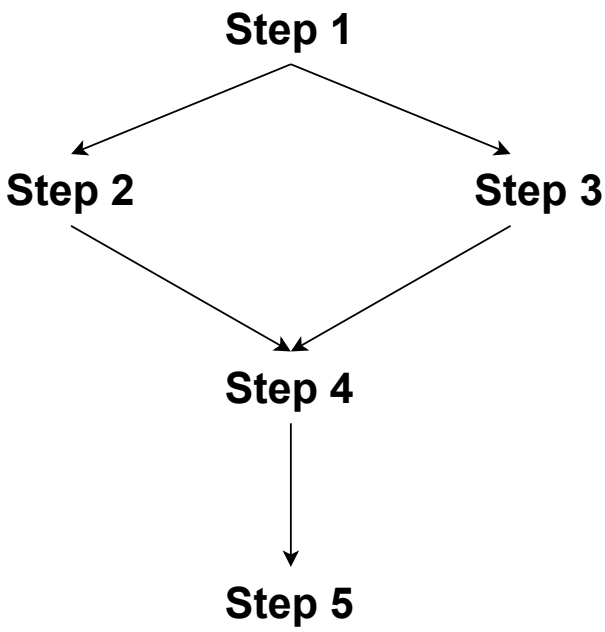
work Flow

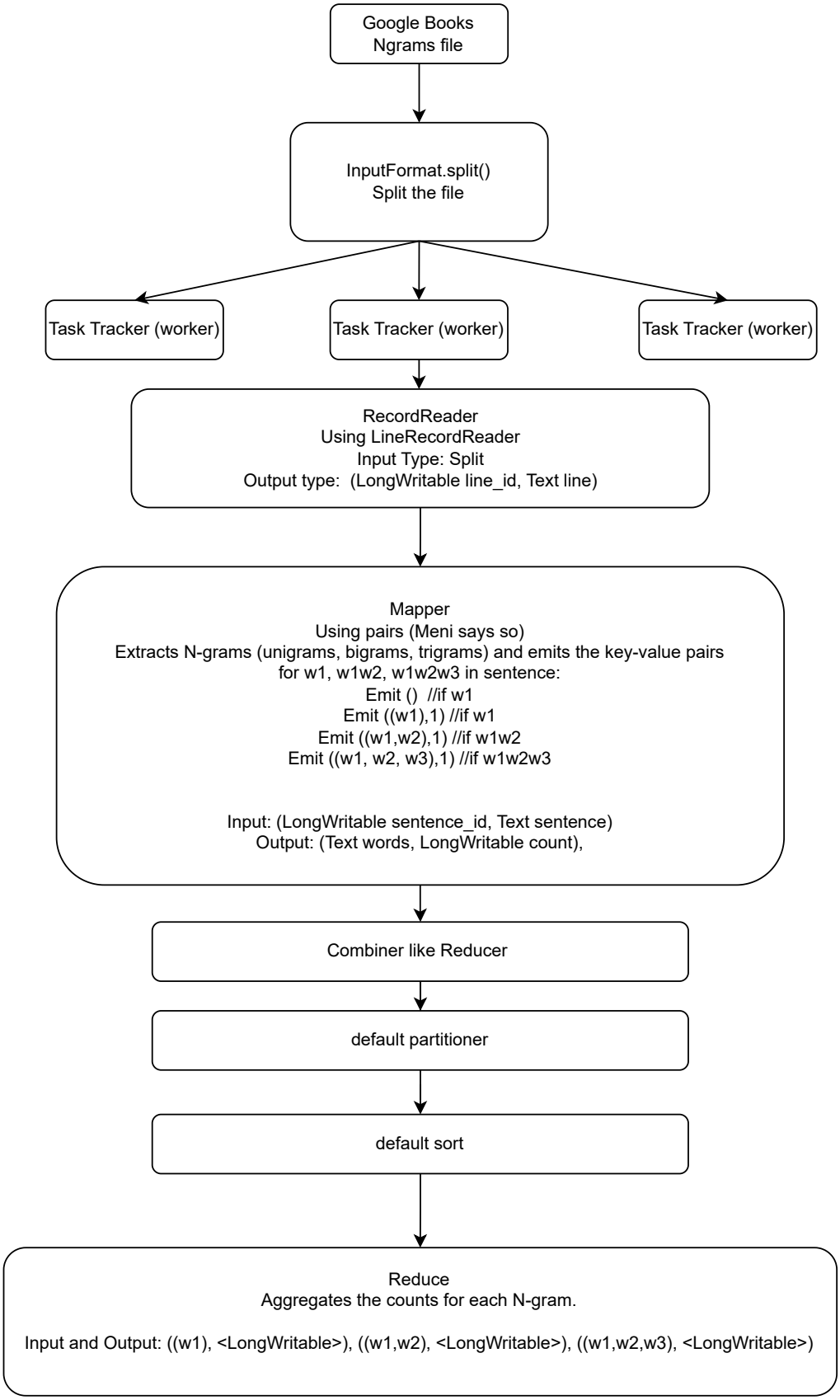


Note: For simplicity, we used Text and spaces instead of arrays.

Step 1

Calculate the number single (w1), pairs (w1,w2) and trio (w1,w2,w3) in the corpus.

w is type <Test>
Input: split from file
Output: ((w1), <LongWritable>), ((w1,w2), <LongWritable>), ((w1,w2,w3), <LongWritable>)



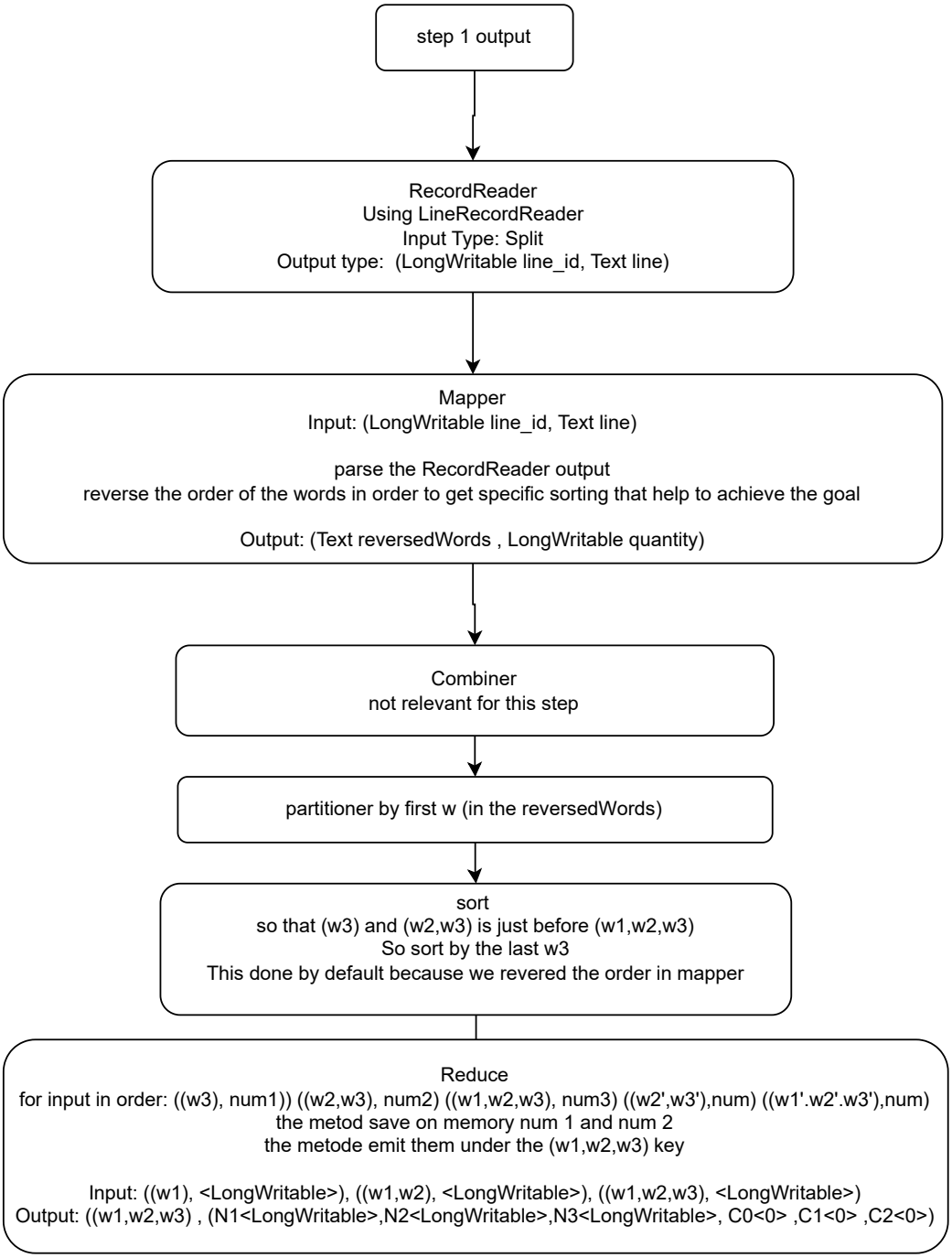
After step 1, we have the number of every single, pair and triple of words

Step 2

Take step1's output as input

Calculate N1, N2, N3 and Emit under the suitable (w1,w2,w3) key

w is type Test
Input: ((w1), <LongWritable>), ((w1,w2), <LongWritable>), ((w1,w2,w3), <LongWritable>)
Output: ((w1,w2,w3) , (N1<LongWritable>,N2<LongWritable>,N3<LongWritable>, C0<0> ,C1<0> ,C2<0>)



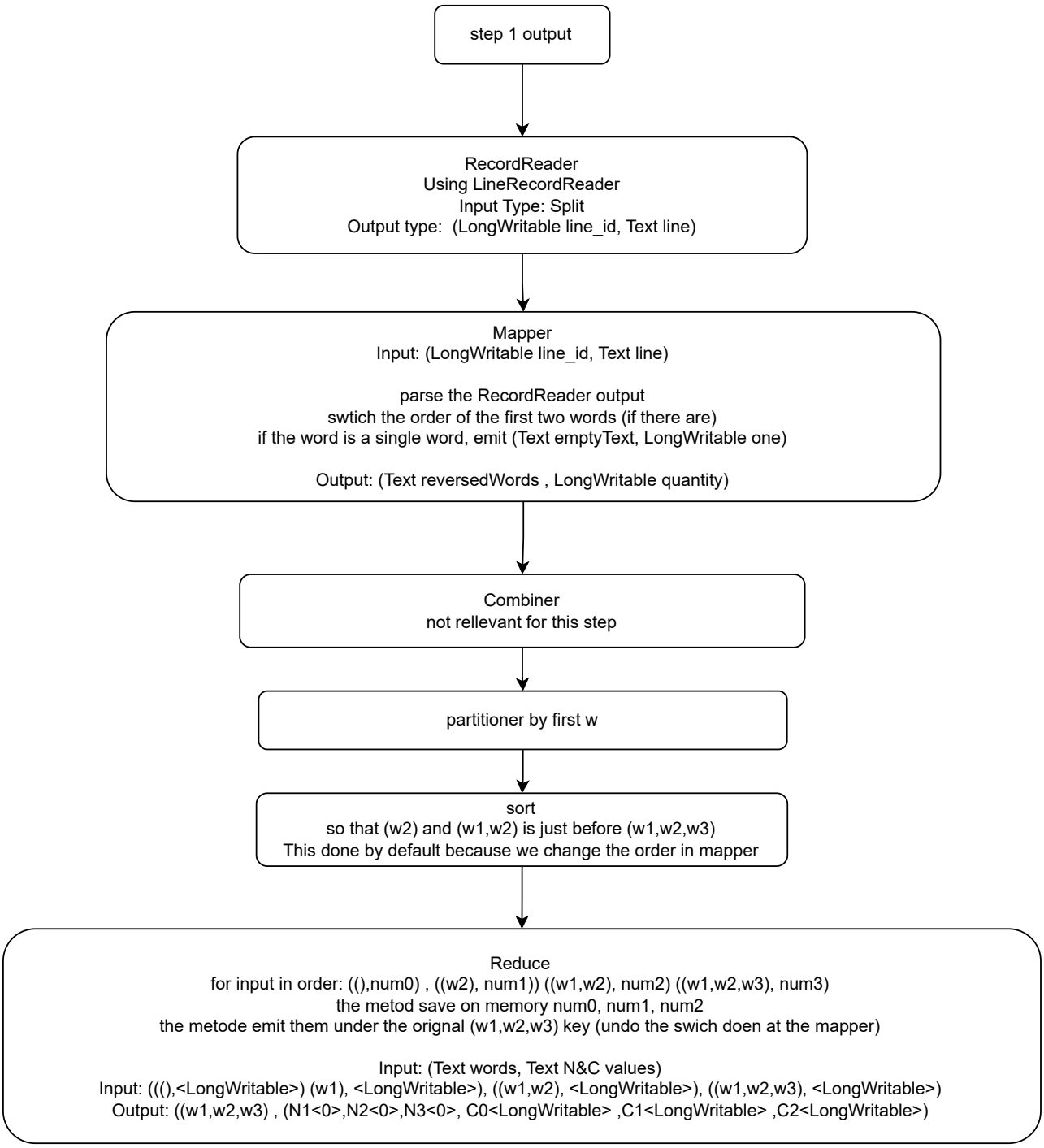
Now we have N1, N2, N3 under suitable (w1,w2,w3) key

Step 3

Take step1, output as input

Calculate C0, C1, C2 and Emit under the suitable (w1,w2,w3) key

w is type <Test>
Input: ((w1), <LongWritable>), ((w1,w2), <LongWritable>), ((w1,w2,w3), <LongWritable>)
Output: ((w1,w2,w3) , ((N1<0> ,N2<0> ,N3<0> , C0<LongWritable> , C1<LongWritable> , C2<LongWritable>))



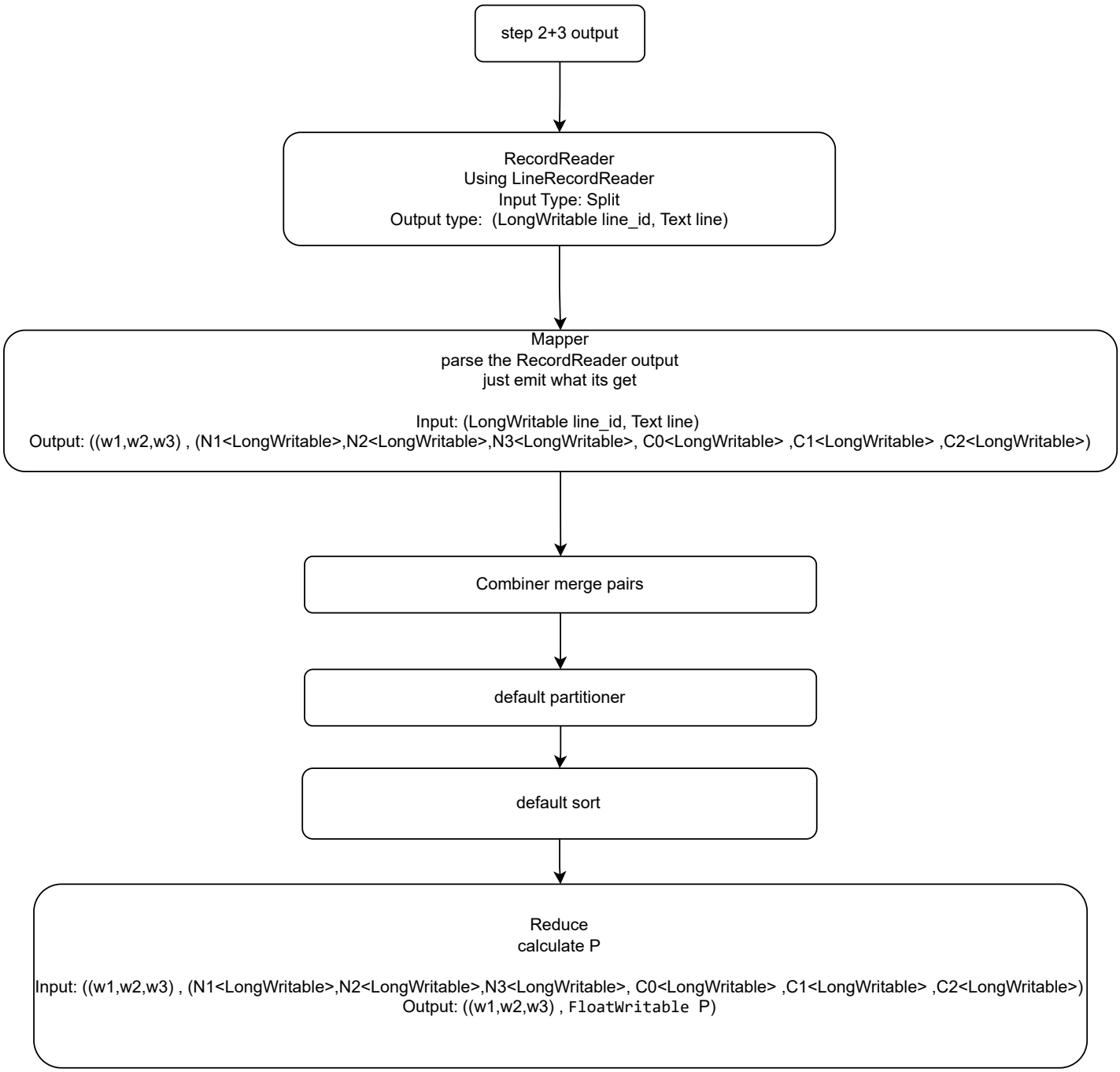
Now we have C0, C1, C2 under suitable (w1,w2,w3) key

Step 4

Take step1+step2 (in the same folder) output as input

Calculate P for each (w1,w2,w3)

w is type <Test>
Input: ((w1,w2,w3) , (N1<LongWritable>,N2<LongWritable>,N3<LongWritable>, C0<LongWritable> ,C1<LongWritable> ,C2<LongWritable>)
Output: ((w1,w2,w3) ,P<LongWritable>)



Now we have P under suitable (w1,w2,w3) key

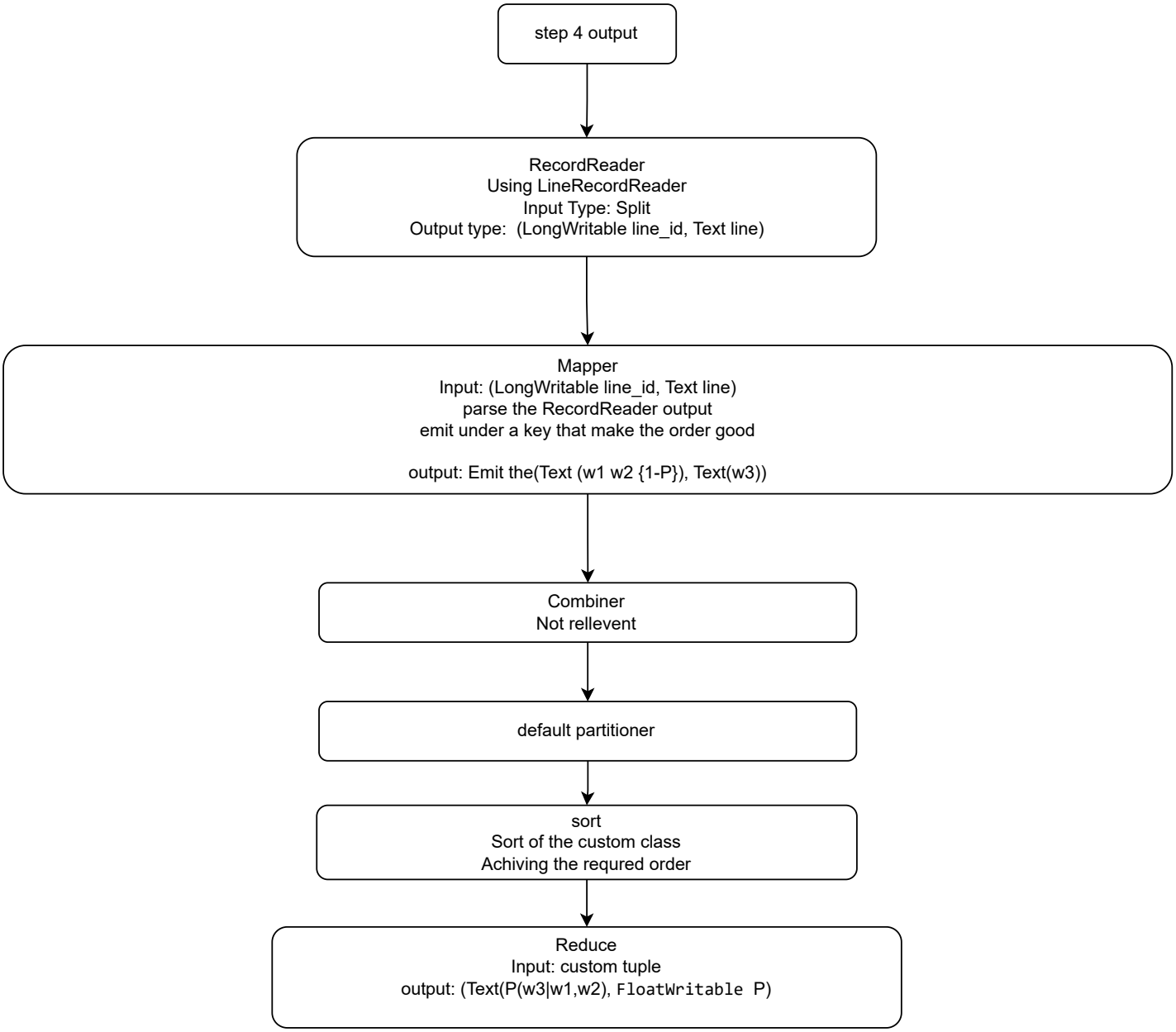
Step 5

Take step4's output as input

The output of the system is a list of word trigrams (w1,w2,w3) and their conditional probabilities (P(w3|w1,w2))). The list should be ordered: (1) by w1w2, ascending; (2) by the probability for w3, descending.

Input: ((w1,w2,w3) ,P<`FloatWritable`>)

Output: (Text (w1,w2,w3) ,`FloatWritable` P)



Finish