Optimization

I take some steps to optimize the "*SetSimJoin.scala"* file.

1. Split the input file which is from the command argument with "\n" to lines.
2. Broadcast the similarity threshold and the "sortValueListBuffer" to reduce the processing time.
3. Increase the partitions, set the "minPartitions = 36".
4. Set the token type to "int".
5. Use "reduceByKey" remove the duplicate.
6. Combine the position filtering with prefix-filtering in PPjoin.
7. Use Jaccard Similarity formula

$$Sim(x, y) = |x \cap y| / |x \cup y|$$

   to match the candidates.
8. Replace the "ListBuffer" type with "List" to reduce the processing time.
9. Use the prefix length formula

$$prefixLen = |x| - \lceil t * |x| \rceil + 1$$

   to find the longest possible prefixes of each record.

At first, since I do not use the optimization method, when using the large case (flickr_london.txt), the program output basically cannot be obtained. For the small case (flickr_small.txt), it takes about 1 minute. After using

these methods, for large case, it takes about 13 minutes, which is much faster than before.