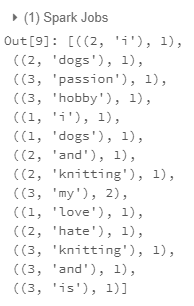
# (ID,texts) -> ((ID,token),1) -> ((ID,token),[1+1+1+...])

map1=line.flatMap(lambda x: [((x[0],i),1) for i in

x[1].split()]).updateStateByKey(updateWC)

reduce=map1.reduceByKey(lambda x,y:x+y)

Text

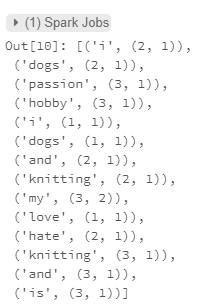
Description automatically generated

# ((ID,token),TF) -> (token,(ID,TF))

tf=map1.map(lambda x: (x[0][1],(x[0][0],x[1])))

tf=reduce.map(lambda x: (x[0][1],(x[0][0],x[1])))

Text

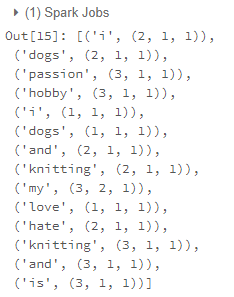
Description automatically generated

# ((ID,token),TF) -> (token (ID ,TF ,1))

map3=map1.map(lambda x: (x[0][1],(x[0][0],x[1],1)))

map3=reduce.map(lambda x: (x[0][1],(x[0][0],x[1],1)))

Text

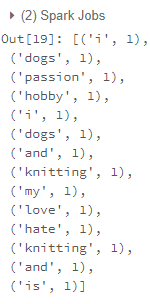
Description automatically generated

# (token(ID,TF,1)) -> (token,1)

map4=map3.map(lambda x:(x[0],x[1][2]))

map4=map3.map(lambda x:(x[0],x[1][2]))

Text

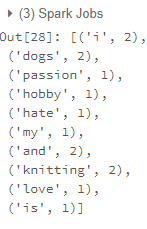
Description automatically generated

# (token,1) -> (token,[1+1+1+...])

reduce2=map4.reduceByKey(lambda x,y:x+y)

reduce2=map4.reduceByKey(lambda x,y:x+y)

Text

Description automatically generated

# (token,#Doc contain word) + (token,max ID) = (token,(#Doc contain word,max ID))

reduce2\_maxID = reduce2.join(ID\_token)

Text

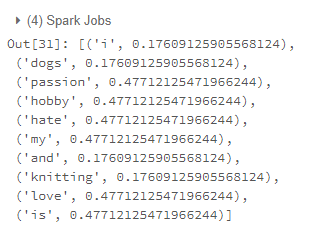
Description automatically generated

# (token,(#Doc contain word,max ID)) -> (token,IDF)

idf=reduce2\_maxID.map(lambda x: (x[0],math.log10(x[1][1]/x[1][0])))

idf=reduce2.map(lambda x: (x[0],math.log10(len(data)/x[1])))

Text

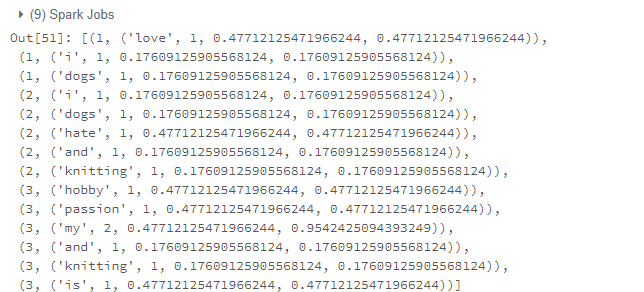
Description automatically generated

# (token,(ID,TF),IDF) -> (ID,(token,TF,IDF,TF-IDF))

rdd=tf.join(idf).map(lambda x: (x[1][0][0],(x[0],x[1][0][1],x[1][1],x[1][0][1]\*x[1][1])))

rdd=tf.join(idf)

rdd=rdd.map(lambda x: (x[1][0][0],(x[0],x[1][0][1],x[1][1],x[1][0][1]\*x[1][1]))).sortByKey()



Text

Description automatically generated