

Patent
patent_number: String
abs: String
claims: String
description: String
title: String
firstName: String
lastName: String
assignee: String
asgNum: String
lat: String
lng: String
country: String
ID: String
td_abs: DoubleMatrix2D
td_claims: DoubleMatrix2D
td_des: DoubleMatrix2D
td_title: DoubleMatrix2D

ParameterLearning
patents: ArrayList<patent>
patentsID: ArrayList<String>
threshold: double
estimateDistanceFunction(): AbstractDistance
generateDistanceFunction(): AbstractDistance

LRWithBoldDriver
training: ArrayList<patent>
trainingID: ArrayList<String>
testing: ArrayList<patent>
testingID: ArrayList<String>
estimateDistanceFunction(): CosDistance
seperateDataset(): void

AbstractDistance
abstractCompare: boolean
claimsCompare: boolean
desCompare: boolean
assigneeCompare: boolean
categoryCompare: boolean
locationCompare: boolean
coAuthorCompare: boolean
lastNameCompare: boolean
firstNameCompare: boolean
titleCompare: boolean
pCorrelation: boolean
weightAbstract: double
weightClaims: double
weightDes: double
weightTitle: double
weightAssignee: double
weightCategory: double
weightLocation: double
weightCoAuthor: double
weightLastName: double
weightFirstName: double
distance(patent first,patent second): double
initialOption(): void
compareAssignee(String str1,String str2,String code1,String code2):double
compareCategories(String str1,String str2):double
compareName(String str1,String str2):double
compareCoAuthor(String str1,String str2):double
compareLocation(String country1,String lat1,String lng1,String contry2, String lat2,String long):double

CosDistance
distance(patent first,patent second): CosDistance
cosDistance(DoubleMatrix first,DoubleMatrix second): double

* pass dataset



1



Training



1



1