**Final Preprocessor:(Java File)**

The data is preprocessed to remove unnecessary words

using libraries and functions.

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package finalpreprocess1;

import edu.stanford.nlp.ie.AbstractSequenceClassifier;

import edu.stanford.nlp.ie.crf.CRFClassifier;

import edu.stanford.nlp.ling.CoreAnnotations;

import edu.stanford.nlp.ling.CoreLabel;

import edu.stanford.nlp.pipeline.Annotation;

import edu.stanford.nlp.pipeline.StanfordCoreNLP;

import edu.stanford.nlp.util.CoreMap;

import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.List;

import java.io.\*;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.\*;

import java.util.Iterator;

public class finalpreprocess{

/\*\*

\* @param args the command line arguments

\*/

private final static String URL\_REGEX = "((www\\.[\\s]+)|(https?://[^\\s]+))";

private final static String CONSECUTIVE\_CHARS = "([a-zA-Z])\\1{1,}[\\s]{1,}";

private final static String STARTS\_WITH\_NUMBER = "[0-9]\\s\*(\\w+)";

protected StanfordCoreNLP pipeline;

public finalpreprocess() {

// Create StanfordCoreNLP object properties, with POS tagging

// (required for lemmatization), and lemmatization

Properties props;

props = new Properties();

props.put("annotators", "tokenize, ssplit, pos, lemma");

this.pipeline = new StanfordCoreNLP(props);

}

public String lemmatize(String documentText)

{

List<String> lemmas = new LinkedList<String>();

// Create an empty Annotation just with the given text

Annotation document = new Annotation(documentText);

// run all Annotators on this text

this.pipeline.annotate(document);

// Iterate over all of the sentences found

List<CoreMap> sentences = document.get(CoreAnnotations.SentencesAnnotation.class);

for(CoreMap sentence: sentences) {

// Iterate over all tokens in a sentence

for (CoreLabel token: sentence.get(CoreAnnotations.TokensAnnotation.class)) {

// Retrieve and add the lemma for each word into the

// list of lemmas

lemmas.add(token.get(CoreAnnotations.LemmaAnnotation.class));

}

}

//Returning string instead of List<String>

String text="";

for(int i=0;i<lemmas.size();i++)

{

if(i==0)

text=text+lemmas.get(i);//first word alone

else

text=text+" "+lemmas.get(i);

}

text=text.trim();

return text;

}

public String data\_collect(String fb\_id)

{

String fb="";

//System.out.println("Data Collection");

try {

URL google = new URL("https://fb.com/anyuser/status/"+fb\_id);

BufferedReader in = new BufferedReader(new InputStreamReader(google.openStream()));

String inputLine;

while ((inputLine = in.readLine()) != null) {

// Process each line.

Pattern pattern = Pattern.compile("<title>([\\s\\S]\*)</title>");

Matcher matcher = pattern.matcher(inputLine);

if(matcher.find()) {

// Rohit Jain observation

fb=matcher.group(1);

break;

}

}

in.close();

} catch (MalformedURLException me) {

System.out.println(me);

} catch (IOException ioe) {

System.out.println(ioe);

}

// fb=fb.replaceAll("on Twitter:"," ");

// fb=fb.replaceAll("&quot;"," ");//usefull

return fb;

}

public static String names\_removal(AbstractSequenceClassifier classifier,String fb) throws IOException {

String s3=classifier.classifyToString(fb,"inlineXML",true);

Pattern pattern = Pattern.compile("<PERSON>([a-zA-Z0-9-\_:., ]\*)</PERSON>");

Matcher matcher = pattern.matcher(s3);

ArrayList<String> names=new ArrayList<String>();

while (matcher.find()) {

// Rohit Jain observation

names.add(matcher.group(1));

}

//remove names from fb

for(int i=0;i<names.size();i++)

{

fb = fb.replaceAll(names.get(i)," ");

}

return fb;

}

/\*

public static void preprocess1() throws FileNotFoundException, IOException

{

try {

FileReader reader = new FileReader("E:/project/ps.txt");

BufferedReader bufferedReader = new BufferedReader(reader);

FileWriter writer = new FileWriter("E:/project/preprocess/p1-ps.txt", true);

BufferedWriter bufferedWriter = new BufferedWriter(writer);

String fb="";

int i=0;

while ((fb = bufferedReader.readLine()) != null)

{

//Remove RT

fb=fb.replaceFirst("RT "," ");

//System.out.println("Tweet 1 is "+fb);

// Remove @username =

fb = fb.replaceAll("@([^\\s]+)", "NAME");

//System.out.println("Tweet 2 is "+fb);

//Remove URL

fb = fb.replaceAll(URL\_REGEX,"hyperlink");

//System.out.println("Tweet 3 is "+fb);

bufferedWriter.write(fb);

bufferedWriter.newLine();

System.out.println(i++);

}

reader.close();

bufferedReader.close();

bufferedWriter.close();

writer.close();

} catch (IOException e) {

e.printStackTrace();

}

}

public static void preprocess2() throws FileNotFoundException, IOException

{

String serializedClassifier = "E:/project/Libraries/stanford-ner-2017-06-09/stanford-ner-2017-06-09/classifiers/english.all.3class.distsim.crf.ser/english.all.3class.distsim.crf.ser";

AbstractSequenceClassifier classifier = CRFClassifier.getClassifierNoExceptions(serializedClassifier);

try {

FileReader reader = new FileReader("E:/project/New Folder/preprocessdata/p1-finaldata.txt");

BufferedReader bufferedReader = new BufferedReader(reader);

FileWriter writer = new FileWriter("E:/project/New Folder/preprocessdata/p2-finaldata.txt", true);

BufferedWriter bufferedWriter = new BufferedWriter(writer);

String fb="";

int i=0;

while ((fb = bufferedReader.readLine()) != null)

{

//Name Removal

fb=names\_removal(classifier,fb);

//System.out.println("Tweet 4 is "+fb);

bufferedWriter.write(fb);

bufferedWriter.newLine();

System.out.println(i++);

}

reader.close();

bufferedReader.close();

bufferedWriter.close();

writer.close();

} catch (IOException e) {

e.printStackTrace();

}

}\*/

public static void preprocess1() throws FileNotFoundException, IOException

{

try {

FileReader reader = new FileReader("E:/project/preprocess4/u1.txt");

BufferedReader bufferedReader = new BufferedReader(reader);

FileWriter writer = new FileWriter("E:/project/preprocess4/pp1/p1.txt", true);

BufferedWriter bufferedWriter = new BufferedWriter(writer);

String fb="";

int i=0;

while ((fb = bufferedReader.readLine()) != null)

{

//Convert all fbs into lowercase

fb=fb.toLowerCase();

//System.out.println("Tweet 5 is "+fb);

// Remove words starting with a number

fb = fb.replaceAll(STARTS\_WITH\_NUMBER,"");

//System.out.println("Tweet 6 is "+fb);

//not correction

fb = fb.replaceAll("n't", " not");

//System.out.println("Tweet 7 is "+fb);

//have correction

fb = fb.replaceAll("'ve", " have ");

//System.out.println("Tweet 8 is "+fb);

//am correction

fb = fb.replaceAll("'m", " am ");

//System.out.println("Tweet 9 is "+fb);

//is correction//ignore has correction

fb = fb.replaceAll("'s", " is ");

//System.out.println("Tweet 10 is "+fb);

//are correction

fb = fb.replaceAll("'re", " are ");

//System.out.println("Tweet 11 is "+fb);

//will correction

fb = fb.replaceAll("'ll", " will ");

//System.out.println("Tweet 12 is "+fb);

//would correction//ignore had correction

fb = fb.replaceAll("'d", " would ");

//System.out.println("Tweet 13 is "+fb);

//us correction

fb =fb.replaceAll("let's", " let us ");

//System.out.println("Tweet 14 is "+fb);

//Remove hash # symbol

// fb = fb.replaceAll("#", "");

//System.out.println("Tweet 15 is "+all\_fbs);

//Remove punctuation mark

fb = fb.replaceAll("\\p{Punct}", " ");

//System.out.println("Tweet 16 is "+fb);

//////emoji,symbols and tamil language removal

fb = fb.replaceAll("[^a-zA-Z0-9 ]", "");

//System.out.println("Tweet 17 is "+fb);

bufferedWriter.write(fb);

bufferedWriter.newLine();

System.out.println(i++);

}

reader.close();

bufferedReader.close();

bufferedWriter.close();

writer.close();

} catch (IOException e) {

e.printStackTrace();

}

}

public static void preprocess2() throws FileNotFoundException, IOException

{

try {

FileReader reader = new FileReader("E:/project/preprocess4/u1.txt");

BufferedReader bufferedReader = new BufferedReader(reader);

FileWriter writer = new FileWriter("E:/project/preprocess4/pp1/pp2/p2.txt", true);

BufferedWriter bufferedWriter = new BufferedWriter(writer);

lemmatization lem = new lemmatization();

String fb="";

int i=0;

while ((fb = bufferedReader.readLine()) != null)

{

//lemmmatization

fb=lem.lemmatize(fb);

//System.out.println("Tweet 16 is "+fb);

bufferedWriter.write(fb);

bufferedWriter.newLine();

System.out.println(i++);

}

reader.close();

bufferedReader.close();

bufferedWriter.close();

writer.close();

} catch (IOException e) {

e.printStackTrace();

}

}

public static void preprocess3() throws FileNotFoundException, IOException

{

try {

FileReader reader = new FileReader("E:/project/preprocess/psdata.txt");

BufferedReader bufferedReader = new BufferedReader(reader);

FileWriter writer = new FileWriter("E:/project/preprocess/p5-first35.txt", true);

BufferedWriter bufferedWriter = new BufferedWriter(writer);

String fb="hi";

int i=0;

while ((fb = bufferedReader.readLine()) != null)

{

// System.out.println("hi "+fb);

if(fb!=null && !fb.equals(""))

{

ProcessBuilder pb = new ProcessBuilder("python","E:/project/netbeans/Tweeter/src/fber/preprocess.py");

Process p = pb.start();

BufferedReader in = new BufferedReader(new InputStreamReader(p.getInputStream()));

fb=in.readLine();

}

else

{

fb ="";

}

// System.out.println("hi "+fb);

bufferedWriter.write(fb);

/\* System.out.println("hi "+fb);

if(fb.isEmpty())

{

bufferedWriter.newLine();

}

else

{

bufferedWriter.write(fb);

}

\*/

bufferedWriter.newLine();

System.out.println(i++);

}

reader.close();

bufferedReader.close();

bufferedWriter.close();

writer.close();

} catch (IOException e) {

e.printStackTrace();

}

}

public static void preprocess4() throws FileNotFoundException, IOException

{

try {

FileReader reader = new FileReader("E:/project/preprocess4/pp1/pp2/pp3/tr25.txt");

BufferedReader bufferedReader = new BufferedReader(reader);

FileWriter writer = new FileWriter("E:/project/preprocess4/pp1/pp2/pp3/pp4/ts25.txt", true);

BufferedWriter bufferedWriter = new BufferedWriter(writer);

String fb="";

int i=0;

while ((fb = bufferedReader.readLine()) != null)

{

fb = fb.replaceAll("([\\s]{2,})", " ");

//System.out.println("Tweet 9 is "+fb);

bufferedWriter.write(fb);

bufferedWriter.newLine();

System.out.println(i++);

}

reader.close();

bufferedReader.close();

bufferedWriter.close();

writer.close();

} catch (IOException e) {

e.printStackTrace();

}

}

public static void preprocess5() throws FileNotFoundException, IOException

{

try {

FileReader reader = new FileReader("E:/project/preprocess4/pp1/pp2/pp3/pp4/ts25.txt");

BufferedReader bufferedReader = new BufferedReader(reader);

FileWriter writer = new FileWriter("E:/project/preprocess4/pp1/pp2/pp3/pp4/pp5/tt25.txt", true);

BufferedWriter bufferedWriter = new BufferedWriter(writer);

String fb="";

int i=0;

while ((fb = bufferedReader.readLine()) != null)

{

//change name to NAME

fb = fb.replaceAll("name", " ");

//System.out.println("Tweet 9 is "+fb);

//change hyperlink to URL

fb = fb.replaceAll("hyperlink", " ");

//System.out.println("Tweet 9 is "+fb);

//Remove additional whitespaces =

fb = fb.replaceAll("([\\s]{2,})", " ");

//System.out.println("Tweet 9 is "+fb);

fb=fb.replaceAll("\\b\\w{1,4}\\b\\s?", "");

bufferedWriter.write(fb);

bufferedWriter.newLine();

System.out.println(i++);

}

reader.close();

bufferedReader.close();

bufferedWriter.close();

writer.close();

} catch (IOException e) {

e.printStackTrace();

}

}

public static void main(String[] args) throws FileNotFoundException, IOException{

// TODO code application logic here

//preprocess1();

//preprocess2();

preprocess4();

// preprocess5();

}//main

}//class