import twitter4j.\*;

import twitter4j.conf.ConfigurationBuilder;

import edu.stanford.nlp.pipeline.\*;

import edu.stanford.nlp.ling.CoreAnnotations;

import edu.stanford.nlp.util.CoreMap;

import java.util.List;

import java.util.Properties;

public class TwitterSentimentAnalysis {

public static void main(String[] args) {

// Twitter API credentials

String consumerKey = "YOUR\_CONSUMER\_KEY";

String consumerSecret = "YOUR\_CONSUMER\_SECRET";

String accessToken = "YOUR\_ACCESS\_TOKEN";

String accessTokenSecret = "YOUR\_ACCESS\_TOKEN\_SECRET";

// Set up Twitter configuration

ConfigurationBuilder cb = new ConfigurationBuilder();

cb.setDebugEnabled(true)

.setOAuthConsumerKey(consumerKey)

.setOAuthConsumerSecret(consumerSecret)

.setOAuthAccessToken(accessToken)

.setOAuthAccessTokenSecret(accessTokenSecret);

TwitterFactory tf = new TwitterFactory(cb.build());

Twitter twitter = tf.getInstance();

// Search query

String query = "Java";

int numberOfTweets = 10;

try {

Query twitterQuery = new Query(query);

twitterQuery.setCount(numberOfTweets);

QueryResult result = twitter.search(twitterQuery);

List<Status> tweets = result.getTweets();

// Stanford NLP Setup

Properties props = new Properties();

props.setProperty("annotators", "tokenize, ssplit, parse, sentiment");

StanfordCoreNLP pipeline = new StanfordCoreNLP(props);

// Analyze sentiments

for (Status tweet : tweets) {

String tweetText = tweet.getText();

System.out.println("Tweet: " + tweetText);

// Perform sentiment analysis

int sentimentScore = analyzeSentiment(tweetText, pipeline);

System.out.println("Sentiment Score: " + sentimentScore);

System.out.println();

}

} catch (TwitterException e) {

e.printStackTrace();

}

}

private static int analyzeSentiment(String text, StanfordCoreNLP pipeline) {

int sentimentScore = 0;

if (text != null && text.length() > 0) {

Annotation annotation = new Annotation(text);

pipeline.annotate(annotation);

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

for (CoreMap sentence : sentences) {

sentimentScore += sentence.get(CoreAnnotations.SentimentClass.class).equals("Positive") ? 1 :

sentence.get(CoreAnnotations.SentimentClass.class).equals("Negative") ? -1 : 0;

}

}

return sentimentScore;

}

}