package com.twitter.search.common.relevance.text;

import java.util.regex.Matcher;

import com.twitter.search.common.relevance.entities.TwitterMessage;

import com.twitter.search.common.util.text.regex.Regex;

public final class LocationUtils {

private LocationUtils() {

}

/\*\*

\* Extract lat/lon information from a twitter message.

\* @param message The twitter message.

\* @return A two-element double array for the lat/lon information.

\*/

public static double[] extractLatLon(TwitterMessage message) {

// first look in text for L:, then fall back to profile

Matcher loc = Regex.LAT\_LON\_LOC\_PATTERN.matcher(message.getText());

if (loc.find() || message.getOrigLocation() != null

&& (loc = Regex.LAT\_LON\_PATTERN.matcher(message.getOrigLocation())).find()) {

final double lat = Double.parseDouble(loc.group(2));

final double lon = Double.parseDouble(loc.group(3));

if (Math.abs(lat) > 90.0) {

throw new NumberFormatException("Latitude cannot exceed +-90 degrees: " + lat);

}

if (Math.abs(lon) > 180.0) {

throw new NumberFormatException("Longitude cannot exceed +-180 degrees: " + lon);

}

// Reject these common "bogus" regions.

if ((lat == 0 && lon == 0) || lat == -1 || lon == -1) {

return null;

}

return new double[]{lat, lon};

}

return null;

}

}