GeoPosition

Plugin: geoPosition

The geoPosition Plugin enables local searches. The plugin parses geographical meta tags (geo.position, DC.coverage.spatial and ICBM). If there are no coordinates in the document, coordinates can be loaded from conf/geodata.txt file.

Download at http://nutch.eventax.com/

Query Syntax

hostel de:70174 (70174 = German zip code of Stuttgart)

hostel de:70174r50

restaurant position:n52e10.0r10

hotel position:s-52e10.0r10

party position:n52.023w10.0r100

politics position:n52e10.0r100

Possible identifiers are n, s, e, w and r.

Use either n or s. Values between -90 degrees and +90 degrees are useful.

Use either e or w. Values between -180 degrees and +180 degrees are useful.

Use r in kilometers. Values bigger then 3000 km might not work.

If the data of other countries is supplied to the system each ISO3166 code followed by colon and the local zip code should bring you to search results.

Config File Options

geoPosition.step

The accuracy positions are stored can be changed in the config file. Default is 1000m.

geoPosition.Domain2PositionFile

Default filename: conf/geodata.txt

The file consists of: URL-Prefix, North and East, tab-separated. Use negative values for south and west coordinates.

Example:

http://www.berlin.de 52.1234 9.9876

http://www.germany.de/berlin 52.1234 9.9876

geoposition.zips.dir

Default dir: zip/

The directory is within the conf dir and should contain files with the center position of each zip of a country. The files for Germany should be called de.geo.txt . The ISO3166 code should be used for the first part of the filename, followed by .geo.txt . You could simply add a file for each country.

geoposition.zips.use

ISO3166 codes from countries which should be used while searching, seperated by semicolon.

Internal Documentation

Our Earth

The plugin assumes that the earth is a globe with 6367km in radius. Calculations get a maximum error of around 0.3%.

The plugin further assumes that that the sea level is the same throughout the whole world. This should not make the error significantly larger.

Coordinate system

To avoid cpu-consuming calculations of sine, cosine and tangent, all the geographic coordinates are transformed before indexing to a 3-D-System with x, y, z.

The Point of origin is the middle of the earth.

x is the line through the Greenwich meridian and the equator.

z is the line through the north pole.

y is 90 degrees to x and 90 degrees to z.

Storeing

The coordinates are stored and not indexed in their polar version (north, east). The coordinates are unstored but indexed in their cartesian version (posX, posY, posZ). If available, the elevation about sea level is stored and indexed in meters (elevation).

Searching

Searches are done by putting a cube around the point of search and searching all stuff between min and max values in each direction. In a 2D view this means, all stuff within a square instead of a circle is fetched. This means a maximum fault of nearly 40% in the distance, hits retrieved from. In the area the fault is around 27%. But it is fast. Distance ranking should minimize this problem in future.

Running search engines using this plugin

- This plugin is used for local searches at http://www.umkreisfinder.de/.

Installation

- Copy the plugin to your crawler and tomcat dir.
- Activate plugin in nutch.conf
- Add the parameters to nutch-conf and the files you need for your country.
- fetch / parse / index
- Then you must see north / east values in your index if you click on explain.
- Try to search

To do

- Checking the existing implementation.
- Speeding up all the stuff
  - Caching Range Queries
  - Using Field Cache
  - Using Hit Collector http://jakarta.apache.org/lucene/docs/api/org/apache/lucene/search/HitCollector.html
- Also using distance for ranking.
- Using whois informations to get positions, if useful.
- Parsing addresses on websites to get positions.

– MatthiasJaekle - 14 Oct 2005