Yahoo! Geocoder

Description

This algorithm converts place names or addresses into Latitude, Longitude co-ordinates. It accepts international addresses, countries, States of United States of America and ZIP codes of United States of America. All co-ordinates are obtained by querying Yahoo! PlaceFinder service. Internet access must be available during geocoding.

Pros & Cons

1. The performance is slower than the Geocoder and may vary due to the network latency since the queries are requested through internet service. The benchmark test geocoded 470 unique locations per minute
2. Yahoo! Geocoder supports address geocoding with international coverage which is not supported by Geocoder.
3. To use Yahoo! Geocoder, user has to obtain an application id through Yahoo! registration. Save your application id and provide it when requested by the Yahoo! Geocoder. Since each application id is allowed to geocode 50,000 locations per 24 hours, the user is encouraged to test on a small set of data first.

Applications

The plugin is useful for scientists who would like to visualize their data on a geographical map (see Geospatial Visualization). User can obtain the geographical coordinates (Latitude and Longitude values) and feed them to the visualization plugin.

Implementation Details

The algorithm receives a list of input data (locations) and queries their locations one by one through Yahoo! PlaceFinder geocoding service. The results will temporarily be cached in memory so that the same query for duplicated locations can be avoided. The cache is deleted after each user request is completed. This plugin is included with the Sci2 application. Performance of this algorithm is O(n).

The detail of the algorithm is shown as following,

1. Yahoo! Gecoder is favored by MVCs (Model-View-Controller)
2. View
   - *YahooGeocoderFactory* is extending *AbstractGeocoderFactory*. It defines all the display options and user input wrapping.. The detail of the implementation can be accessed through here.
3. Controller
   - *GeocoderAlgorithm* is the shared geocoder controller which is documented with *AbstarctGeocoderFactory*. It invoked *YahooFamilyOfGeocoder* methods to retrieve the geocoder based on the selected type and performs the geocoding operation.
   - *YahooFamilyOfGeocoder* contains four geocoders: *YahooCountryCoder*, *YahooStateCoder*, *YahooZipCodeCoder*, *YahooAddressCoder*. Each coder will only be created when it is invoked
   - *Geocoder* contains the geocodingFullForm and geocodingAbbreviation methods which will invoked *PlaceFinderClient* to request Yahoo! PlaceFinder service.
   - *PlaceFinderClient* will performs the service query to Yahoo! PlaceFinder. Every request has three times retry for network failure. The result will be returned as ResultSet which is defined in the model
4. Model
   - The model classes were generated by using placeFinder.xsd and located in *edu.iu.scipolicy.preprocessing.geocoder.coders.yahoo.placefinder.beans*.
   - The JAXB technology is chosen as the un-marshaller since Yahoo! PlaceFinder have a simple and standard service response definition at here. JAXP is more suitable for complicated data model which provides more control in data preprocessing.
5. Dependency: javax.xml.*

Usage Hints

Here is a 7 steps guide to using the plugin:

1. Make sure you are connected to the internet.
2. Load an input data table that contains locations to be geocoded.
3. Select **Analysis > Geospatial > Yahoo! Geocoder** from the menu bar. A window will pop up.
4. Enter your application id. You can obtain one from here.
5. Choose place type that represents your input location data. The place type can be address, country, U.S. state or U.S. ZIP code.
6. Choose place name column that represents the location field in your data file.
7. Select Include address details if you want Yahoo! to return the parsed address information.
8. Press Ok to start geocoding.

All rows of the data will be geocoded one by one using Yahoo! PlaceFinder. Emtpy entries and invalid locations that failed to be geocoded are listed in the console.

The output of this algorithm is the original input table with two additional for latitude and longitude. Locations that failed to be geocoded will have blank entries.

Performance varies by machine and network latency. Our benchmark is 470 unique locations per minutes.

Links

- Source Code: Link
- Root Source: Link
- External Package: Link
- Yahoo! PlaceFinder: Link

Acknowledgments