

## P-A-41 オフラインによるジオコーディングの Python による実装

田渕 健 東京都立駒込病院小児科・東京都がん登録室(兼務)

**【目的】** がん登録業務において地理情報の正規化は照合精度の向上に寄与し、また地理情報の決定はがん登録情報の利活用に重要なプロセスである。ジオコーディングには、通常 GIS ソフトウェアを用いるが、商用の ArcGIS は高機能だがきわめて高価であり、QGIS などのフリーソフトウェアも含め、活用には熟練を要する。しかも、細かな処理は Python 等のプログラミング併用が必要であり、更に API を通じて地理情報の外部ノード情報の参照処理を行うこともありうる。そこで、必要な機能のみを大量高速処理を実現し、しかも、完全オフラインで運用できる手法を Python によってのみ実装したので報告する。

**【方法】** 国土交通省が無料で提供している位置参照情報ダウンロードサービス<sup>1)</sup>には、**大字・町丁目位置情報**(住居表示地区では丁目迄、約 19 万件弱)と**街区位置情報**(街区番号・地番迄、1,964 万件強)の 2 通りがある。目的とする住所情報の精度に応じてこれらの 2 種類の精度の位置情報と照合を行い、緯度・経度を決定するというのが目標となる。残念ながら、日本における住所の表記やコード化が、様々な住所データベース間で齟齬があるのが実情であり、国管理の全国がん登録の住所コード化に用いている地方公共団体情報システム機構(J-LIS)の全国町・字ファイル(66 万件強)とも異なっている。このため、まず、①**住所情報間の相互参照が可能となるリンクを作成することが必要である。**この作業の要は**住所正規化の実装**である。また街区位置情報を利用するには、個々の住所における街区番号・地番までの住所正規化が必要である上、照合処理の負荷は大きいので、プログラミング処理の活用と共に、②**大字・町丁目位置情報の実用上の精度の検討**が必要である。即ち、街区位置情報と対応する大字・町丁目位置情報の緯度・経度に基づく距離の差を評価し、町字レベルでの位置情報で代用出来るか検討する。距離の計算には**測地線距離**(回転楕円体)を用いた<sup>2,4)</sup>。これらの処理は Python3.7 によって実装した。主なパッケージは、geopy<sup>5)</sup>(ただし API 機能は用いず、2 点間距離算出機能のみ使用)と folium<sup>6)</sup>(地図描画とプロットやコプロスマップの作図)を利用した。ソフトウェアやオープンデータ等のダウンロードを除き、プログラミングツールのインストールやデータ処理は、完全なオフライン環境で作業を行った。

**【結果】** 住所正規化は、まず表記住所から都道府県、市区郡町村、町字、街区番地を抽出する。合併等による市区町村名変更や住居表示変更、よくある誤表記は、事前リストを作成し、一斉に修正する。数字表記のゆれ(アラビア数字と漢数字の混在)、「がのつ」の表記のゆれ、丁目・番・号のハイフン表記、都道府県名の省略の補完等、一連の住所正規化作業を行う。多くはロジックレベルでの自動処理を行うが、住居表示変更、誤字脱字などの非定型処理は手動で修正が必要となるものもある。登録データで住所が町字レベルまで記載されているものの内 99%以上が自動処理でジオコーディング可能であった。決定不能例は位置参照情報サービスに対応するものがないものと直接番地表記や小字などの住所表記が一意的でないものであった。

緯度経度表現の町字単位と街区単位の緯度経度による位置情報間の距離を算出すると、有効件数約 1960 万件の平均は 0.68km であった。町字単位の平均値と比較すると、街区位置情報の 95%は、対応する町字位置情報の 1km 未満である。これを市区町村別平均で見ると、町字位置・街区位置間距離は、東京都 23 区では、町字位置と街区位置の差が 0.316(=0.1<sup>0.5</sup>)km 以下であった。

**【考察】** ジオコーディングを行うためには事前準備として住所正規化は不可欠であり、住所正規化精度によってジオコーディングの精度が決まる。国土交通省が提供する位置情報参照サービスの精度は街区レベル迄可能であるが、全国町・字ファイルと位置参照情報ダウンロードサービスとでは情報更新時期が異なっており、後者はややバージョンが古いため対応するものがなく緯度経度が検索できない場合がある。町・字ファイルは診断日に合わせた地名の履歴情報利用が可能であるが、位置参照情報ダウンロードサービスは更新時期の分のみであるため、最新住所に変換する必要がある。都心のように地域によっては町字レベルのジオコーディングのみで十分精度の高い位置情報が得られる。geopy による 2 点間測地線距離計算は十分に高速処理が可能であった。

**【参考文献】**

- 1) 位置参照情報ダウンロードサービス <http://nlftp.mlit.go.jp/isi/>
- 2) 三浦英俊: 緯度経度を用いた 3 つの距離計算方法. オペレーションズ・リサーチ, 2015.
- 3) 長坂直彦: 回転楕円体上の測地線及び航程線の算出について. 海洋情報部研究報告 2013.
- 4) Wei-Kuo Tseng, Jiunn-Liang Guo, and Chung-Ping Liu: A Comparison of Great Circle, Great Ellipse, and Geodesic Sailing. Journal of Marine Science and Technology, 2013.
- 5) GeoPy Contributors: GeoPy Documentation Release 1.20.0. 2019.
- 6) Folium 0.10.0 documentation <https://python-visualization.github.io/folium/#>

