An Examination of Geocoder Strategies Given a Large Sample of Correct Addresses

based on PostGIS Tiger Geocoder revision 7687, July 2011

Brian Hamlin, maplabs@light42.com; Bob Spence bgspence@gmail.com

Sample Dataset

Two hundred thousand residential addresses, directly from the rolls of the Alameda County Assessor's Office. Alameda County is the sixth most populous county in California, and roughly 18th in the nation with 1.8 million residents listed in the year 2000. Since the sample addresses are directly from the property records of the county, the addresses are by one definition, 100% accurate and correct mailing addresses.

Introduction

One reason for the success of the World Wide Web was that it was widely understood from the beginning that since web sites would generally be created by both technical and nontechnical users, interpretation of the contents of a web site, in markup language, would best be understood using mechanisms that allowed for mistakes and inconsistencies. The machine interpretors of web site content, browsers, were built from the beginning to allow for error in markup, to recover from reading errors gracefully, and to give acceptable results. Notwithstanding errors in markup, correct markup was certainly supposed to be handled without ambiguity.

Reading web content markup was non-trivial, and quickly became more so over time.. yet the balance of correct output for correct input, and acceptable output for problematic input, continued to be kept as web browsers evolved.

In the case of a geocoding engine, it is desirable the a similar balance be struck. Addresses to be geocoded will originate from sources of varying quality, accuracy and correctness. Therefore, it is desirable to expect the geocoding engine to gracefully accommodate error, ambiguity and contradiction, but without losing sight of accurate results given accurate input.

Two Cases

The parsing of an address is non-trivial – perfection is arguably unattainable. Sources of error and ambiguity in the geocoding process can originate in the address input, or in its interpretation, or both. It would be wise to acknowledge the limitations of the parsing process, while at the same time refining strategies for better handling of both flawed inputs and correct inputs.

In this test case, the addresses fed as input are not flawed, in fact, they embody a definition of correct input. The current behavior of the geocoder gives results with a huge variation in quality. What strategies might be embraced moving forward as the geocoding engine is refined?

Hints and their Significance

All pieces of the address input could be thought of as hints. Correlations in those hints ought to be considered as strong indicators, whereas lone aberrations or lack of clarity in one hint ought to be given lower weighting.

Let us consider some basic pieces of a mailing address, and their relative “messiness.”

State	not messy; unambiguously defined boundaries and symbolic representations
Zip Code	not messy; ZCTAs in the TIGER data are well defined spatially, and a direct match ought to have strong weighting.

Note that it appears that zip codes have some built in error correction – a zip code digit that is off-by-one is **nowhere near** its alpha-numeric neighbor. Many simple errors in zip appear to change the US State the zip code is in, making zip+state correlation a very strong hint

Cities / Towns messy; There seem to be many kinds of settlements that are referred to in colloquial terms, in mail addresses, that are not in TIGER place names, and other problematic cases; perhaps more so in older parts of the country.

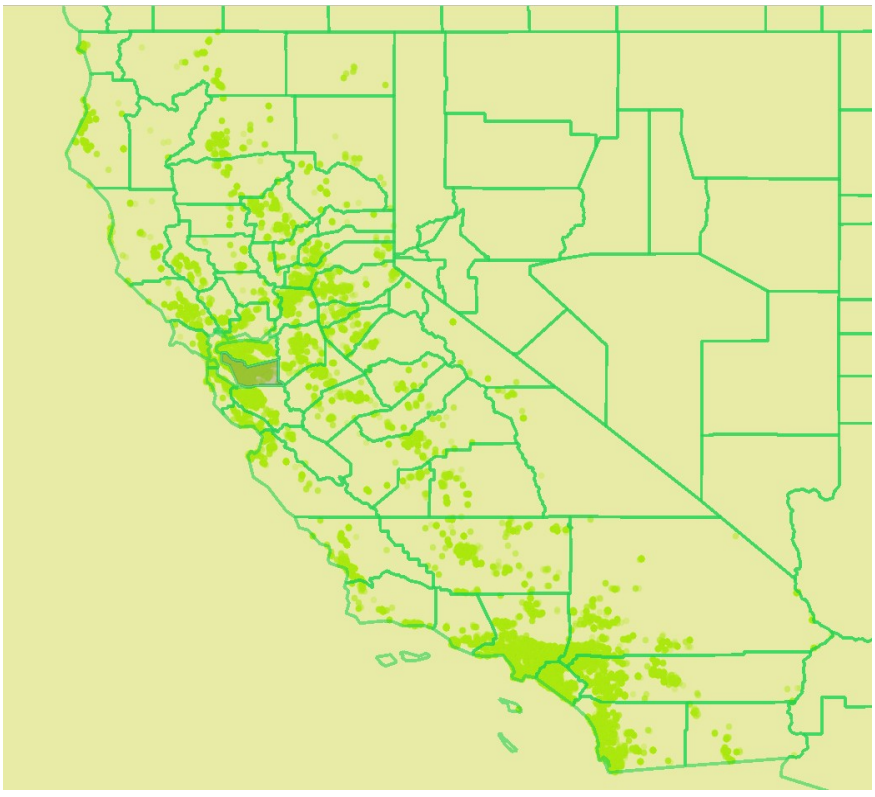
Address very messy; the most difficult to parse, error prone on the part of the supplier, known to be implemented in the geocoding engine at a "best-effort" level of completeness, and as said, the handling of which will never be flawless, in practice

Tail Wagging the Dog

There is folk wisdom embodied in this phrase – don't let an exception take precedence over the majority of cases, and if it does, you have a problem. Here is a graphic of the results of geocoding the sample Alameda addresses, using trunk geocoder rev 7687

Geocoding results from 200k correct addresses in Alameda County (marked in gray)

All alternative results are shown



It is worth emphasizing, every address in this sample had a correct City/Zipcode combination, and a correct street name and street number. It is straightforward to machine comprehend the zip code and state, and in this case, the vast majority of city names were correct in a straightforward way. Rather than take this opportunity to lament the execution of the geocoding engine, let's take the opportunity to think about what worthwhile goals and strategy for the geocoding engine might be in the future.

Locus of Localities

Errors or mismatching pieces of address inputs might be thought of as to each represent a locus of localities. What are the anchors? What are the possibilities given an anchor? If more than one component of an address are in correlation, then what weight might be given to that “hint” ?

Locality can be thought of both geographically and lexically. It is a good thing that the geocoding software is sophisticated enough to be able to make decisions based on lexical locality, most notably the soundex stage of parsing. But it appears that currently, lexical locality is taking precedence over strong hints of geographic locality.

The City plus Zip Hint

If it is the case, that for a given input address, the city portion of the norm_addy struct is an exact match to a known city, and that match is coincident with the provided zip code and state, then what should the output of the geocoding engine be ?

This map shows that currently, the geocoding engine gives results such that, the likelihood of a result is roughly proportional to the number of known addresses in any part of the state. It appears that by fluke or by design, the most “messy” and error prone part of the input address, by both the supplier and by machine interpretation, is searched for first, wherever it may occur, apparently disregarding the zip code and city name.

Given that a zip code can be verified with a city name, that a mistaken zip code appears to almost never occur in the city corresponding to the original zip code, and that State is highly unambiguous, it may be much better design to weight a match of state, zip code and city name highly, and constrain result choices to local choices.

P.S. – A Suggestion

It would be useful to have a geocoder with input data quality input parameters. You could specify types of guessing allowed in the geocoder based on the nature of the input data set. These could be global settings used for an entire input data set.

If you know the addresses are all in California, that would be useful.
If you knew that the city, state and zip should be good you could weight that.
If you knew the address was good, but wanted some guesses to city, that would be good to specify.
If you knew the spelling was suspect, you could specify that, or turn off soundex for good input datasets.

This could be useful in a multi pass geocoding process, Geocode with tight specs first, then loosen them on the misses.