POI Implicit Type Tagging Based on Name

12/14/17

Goals

- Add type information to poorly typed or untyped POI datasets
  - e.g. for a point named “North Central High School” with no type tag present, add tag amenity=school
  - e.g. For “Joe’s Pizza”, add tags amenity=restaurant and cuisine=pizza
- Add as few incorrect type tags as possible
- POIs with existing types should only have new types added that have more detail
  - e.g. If POI has amenity=hall, add amenity=public_hall - but not the other way around
- Tagging rules should be auto-generated but also editable by a human
- The processing time to add the implicit tags to POIs should be comparatively much less than the time it takes to conflate the datasets

Use Cases

- When would I want to use this feature?:
  - You have a large POI dataset containing valuable information to combine with other datasets but its type information is non-existent or vague (lot of just poi=yes tags)
  - You're ok with a small percentage of incorrect type tags being added to your data
- When would I **not** want to use this feature?:
  - You have a smaller POI dataset with poor type information which can be manually edited in a short amount of time
  - You absolutely don't want any incorrect type tags added to your POI data
- Specific Use Cases
  - Wikimapia doesn't have great type attribution in general:
    - OSM/Wikimapia POI conflation
    - GeoNames/Wikimapia POI conflation

Approach

- Examine the entire public OSM and GeoNames datasets:
  - OSM: ~4.1 billion nodes
    - ~25 million POIs
  - GeoNames: ~12 million nodes
    - ~11 million POIs
- Hootenanny defines POIs as any node with a name or a type that falls in its POI schema
- For POIs with a non-generic type (something other than poi=yes) record an association of each of the individual words in the POIs name with any type tags present on the POI
  - e.g. There were ~475,000 times when a POI had “school” in the name and the POI was tagged with amenity=school
- Select a word/tag occurrence minimum threshold below which implicit tags rules are not to be created - So far, ~1000 seems to be a sweet spot.

Approach (continued)

- Attempt to add more detailed type information using these generated associative rules for each POI, as long as the contents of a POIs name don't involve more than one rule; e.g.:
  - "Fairmont Hotel Swimming Pool" - hotel and swimming pool belong to two different rules
  - "Police Hospital" - police station and hospital belong to two different rules
  - "Glen Park Substation" - park and a power substation belong to two different rules
- The above examples are solvable problems (I think) if you start applying some more advanced lexical parsing techniques...but wanted to keep things simple starting out, so simply passing on dealing with those types of situations for now.
- Translate all POI names to English
  - Trying to reduce the size of the rules and make debugging easier
  - Didn't seem to make a big difference in tagging correctness; needs more investigation
  - Hoot's translator isn't great or I'm not using it completely correctly

Approach (continued - 2)

- Add an option to restrict association rules to only words made up of some token in Hootenanny's list of OSM schema type values; e.g.:
    - Name containing "school" would be considered since schema contains amenity=school
    - Name containing "car parts" would be considered since shop=car_parts is in the schema but name containing "auto parts" gets missed (look for synonyms somehow?...complicated)
    - This option reduced a lot of incorrect type tags added initially, but don't want to have to use it since it excludes so many potentially useful tagging rules
- Search for matches in tokenized name word groups of size two or greater before trying to match against individual words
    - e.g. for name="Central Bus Station", also create mappings for "Central Bus" and "Bus Station" in addition to "Central Bus Station", "Central", "Bus", and "Station"
    - Given a POI name, search for a match against "Bus Station" before searching for matches against "Bus" or "Station"
    - Word grouping sizes greater than two have yielded no additional gains so far

Approach (continued - 3)

- Create a custom rules override file to change any rules that don’t seem right
    - Trying to limit having to edit it to avoid manual work, but some changes are necessary
    - e.g. “beach” was very often mapped to tourism=hotel, which is too vague, so made sure “beach” by itself was always ignored

Challenges

- Memory constraints
    - VM’s in VirtualBox don’t release memory after separate processes get run
        - Run hoot, then run Unix sort command, etc.
- Processing time constraints
    - Total rule database generation time: ~6.5 hours
    - Broke that down into separate commands to facilitate testing rule filters more easily:
        - POI filtering time: ~5 hours
        - Raw rule generation time: ~1.5 hours
        - Time to apply filters and generate a rules database: < 1 minute
            - This one can be repeated often to experiment with different filtering options

Challenges (continued)

- A fair amount of incorrect type tags added initially:
  - There was a lot of noise due to the wide variety of word/tag associations across several languages
  - Picking too low of an occurrence threshold increased the noise and greatly increased the processing time
  - Picking too high of an occurrence threshold would miss too many good tagging opportunities

Results (so far)

- Based on 16 regression tests:
  - **For POI to POI conflation datasets where one of the inputs has poor type tagging (Wikimapia), saw conflation incorrect match decreases of 3-4% when implicit tagging was applied, with additional room for improvement**
  - For POI to POI conflation test datasets with generally good tagging, seeing no adverse effects to conflated output and a small number of new correct matches
- Incorrectly added type tags
  - Made some code changes and custom rule additions to reduce incorrectly added tags
  - Dealing with additional incorrect tags added on a case by case basis
  - Will probably never get rid of all of them
- Don't have good numbers available yet on:
  - the ratio of correctly added type tags to incorrectly added tags (false positives)
  - how much processing time tagging adds

Future Work

- Keep making tweaks to weed out more incorrect type tags:
    - Should be an ongoing process (like Hoot schema additions)
    - Try out some more advanced name parsing techniques
    - Want to avoid the tagger having a bunch of very specific, spaghetti-like rules as the code would quickly become unmaintainable; simpler is better
- Make sure Hoot schema is in sync with most or all of the tagging rules
    - Implicit tagging rules will do no good if there aren't corresponding entries in the Hoot schema to be used during conflation (for the most part, they already match up)
- Implicit tagging concept could be extended to data types other than POI (buildings, transportation, etc) - not sure how useful that would end up being
- Add additional input data sources beyond OSM and GeoNames when generating the rules (which ones?)
- Regenerate the rules periodically against the latest open source data