From Label Address to Geocode: Strategies for an Implementation

Candidate: Stefano Forti
Supervisors: Anna Bernasconi | Michela Bertolotto
The Project

• It is related to an actual software design problem encountered within an Irish start-up company, XPRESO.
• Courier deliveries optimization.
• How to build a scalable and efficient geocoder, without relying on proprietary software or data.

• XPRESO team is working on implementing the described system.
Geocoding

• It is the process of assigning geographic coordinates to a piece of information (usually a postal address)
• «Geographic location is a key feature of 80-90% of all government data» [US Federal Geographic Committee, 2003]
• Nowadays, it is part of our daily life
A Toy System

- Input
- Processing Algorithm
- Reference Dataset
- Output
- Clerical Review
- Accurate Matching
- Efficient Retrieval
Processing Algorithm

1. remove the punctuation
2. convert everything into lower or uppercase
3. remove useless information
4. adjust common misspellings for street names word by word
5. standardize common abbreviations
6. tokenize the address
Approximate String Matching

Let $\Sigma$ be a finite alphabet of symbols and $\sigma, \rho \in \Sigma^*$ strings over that alphabet. The string $\sigma$ matches the so-called pattern $\rho$ if $d(\sigma, \rho) < \eta$ where $d: \Sigma \times \Sigma \to [a, b] \subseteq \mathbb{R}$ is a distance function (measuring the similarity between the two strings) and $\eta \in (a, b)$ is a similarity threshold.
Levenshtein Distance

\[ L(i, j) = \begin{cases} 
  i & \text{if } j = 0 \\
  j & \text{if } i = 0 \\
  L(i - 1, j - 1) & \text{if } \sigma_i = \rho_j \\
  \min\{L(i - 1, j), L(i, j - 1)\} + 1 & \text{if } \sigma_i \neq \rho_j 
\end{cases} \]

- aka edit distance
- dynamic programming solution has time complexity \( \mathcal{O}(|\sigma||\rho|) \)
- Actual distance

\[ d_L(\sigma, \rho) = 1.0 - \frac{L(\sigma, \rho)}{\max\{|\sigma|, |\rho|\}} \in [0, 1] \geq \eta = 1 - \delta_2 \]
Length Distance

• Strings with a huge **difference in length** cannot match; e.g.:

10, Leeson Street ≠ 87, Sydney Parade Avenue

• Length Distance, time complexity \(O(1)\):

\[
d_0(\sigma, \rho) = \frac{\text{abs}(|\sigma| - |\rho|)}{\max\{|\sigma|, |\rho|\}} \geq \gamma_0 = 1 - \delta_0 \in \mathbb{R}
\]

(length test)
Bag Distance

- Thinking of $\sigma, \rho$ as the two bags (multisets) $S, R$

$$d_B(\sigma, \rho) \overset{\text{def}}{=} d_{BAG}(S, R) = \max\{|S \setminus R|, |R \setminus S|\}$$

- **lower bound** for edit distance
  - $d_1 = \frac{d_B(\sigma, \rho)}{\max\{|\sigma|, |\rho|\}} \leq \gamma_1 = 1.0 - \delta_1 \in [0,1]$ (bag test)

- Time complexity: $\mathcal{O}(|\sigma| + |\rho|)$

  - e.g. $\sigma = \text{\textquoteleft pearls\textquoteright}$ and $\rho = \text{\textquoteleft teasels\textquoteright}$
  - $S = \{p,e,a,r,l,s\}$ and $R = \{t,e,a,s,e,l,s\}$
  - the distance is $d_B = \max\{|\{p,r\}|, |\{t,e,s\}|\} = \max\{2,3\} = 3$
Let $M$ be the number of entries in the GeoDB; the overall cost is

$$C = M \cdot \text{cost}(d_0) + (p_1 \cdot M) \cdot \text{cost}(d_B) + (p_2 \cdot M) \cdot \text{cost}(d_L)$$

Time savings

$$S \approx \frac{C}{M \cdot \text{cost}(d_L)} = 1 - \frac{\text{cost}(d_0)}{\text{cost}(d_L)} - p_1 \cdot \frac{\text{cost}(d_B)}{\text{cost}(d_L)} - p_2$$

$$= 1 - 0.003 - p_1 \cdot 0.82 - p_2$$
The Strategy

- Data stored in the GeoDB is assumed to be clean, correctly geocoded and verified.
- Data stored in the GeoDB is used to create the gazetteers.
A Worked Out Example

- A dataset of 4,733 geocoded addresses has been used.
- 463 were geocoded and have been used as GeoDB.
- The entire dataset used for queries.
- Theoretical previsions were confirmed.

<table>
<thead>
<tr>
<th>$\delta_0$</th>
<th>Discarded After Length Test</th>
<th>Discarded After Length and Bag Tests</th>
<th>Time Needed</th>
<th>Empirical Savings</th>
<th>Theoretical Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.75</td>
<td>34%</td>
<td>94%</td>
<td>10</td>
<td>38%</td>
<td>42%</td>
</tr>
<tr>
<td>0.80</td>
<td>47%</td>
<td>94%</td>
<td>8</td>
<td>50%</td>
<td>53%</td>
</tr>
<tr>
<td>0.85</td>
<td>59%</td>
<td>94%</td>
<td>7</td>
<td>56%</td>
<td>63%</td>
</tr>
<tr>
<td>0.90</td>
<td>73%</td>
<td>95%</td>
<td>5</td>
<td>68%</td>
<td>74%</td>
</tr>
<tr>
<td>0.95</td>
<td>86%</td>
<td>97%</td>
<td>3</td>
<td>81%</td>
<td>85%</td>
</tr>
</tbody>
</table>
A Worked Out Example (1)

• It was possible to correctly – or nearly correctly – geocode the 45-50% of the data with an average edit distance of 5.9.

• e.g.

One match found for 101,balkill park,howth, ,dublin
It is an almost exact match!
85 balkill park, ,howth, ,dublin ED = 6

• Next step is to adjust the values of $\delta_0$, $\delta_1$ and $\delta_2$ in order to maximize the performance over the complete database.
Pros & Cons

**Pros**
- completely **independent** from external services
- even more **reliable** than those services (doorstep accuracy)

**Cons**
- slow start
- human **intervention** and **decision** required
The GeoDB
La Pinta
10, Highland Grove,
The Park – Cabinteely,
Dublin 18, Dublin, EIRE

- hierarchical structure and
detail level
Conclusions & Future Work

• Guidelines for implementing an efficient, reliable and scalable geocoder
• Simple theoretical model proposed and verified
• Positive feedback from XPRESO

• Future plans would include: testing over the actual dataset and/or integrate probabilistic techniques, based on hidden Markov models.
Questions & Answers

Thank you!