Ad-Hoc Georeferencing of Web-Pages
Using Street-Name Prefix Trees

Andrei Tabarcea¹, Ville Hautamäki², and Pasi Fränti¹

¹ Speech and Image Processing Unit, University of Eastern Finland, Joensuu, Finland
² Human Language Technology Department, Institute for Infocomm Research
A*STAR, Singapore 138632, Singapore
{tabarcea,franti}@cs.joensuu.fi,
vmhautamaki@i2r.a-star.edu.sg

Abstract. A bottleneck of constructing location-based web searches is that most
web-pages do not contain any explicit geocoding such as geotags. Alternative
solution can be based on ad-hoc georeferencing which relies on street
addresses, but the problem is how to extract and validate the address strings
from free-form text. We propose a rule-based pattern matching solution that
detects address-based locations using a gazetteer and street-name prefix trees
created from the gazetteer. We compare this approach against a method that
doesn’t require a gazetteer (a heuristic method that assumes that street-name has
a certain structure) and a method that also uses data structures created from the
gazetteer in the form of street-name arrays. Experiments using our location
based search engine prototype (MOPSI) for Finland and Singapore, show that
the proposed prefix-tree solution is twice as fast and 10% more accurate than its
rule-based alternative and 10 times faster if an array structure is used when
accessing the gazetteer.

Keywords: Search engine, LBS, Database, prefix tree, Georeferencing, Mobile
device, Location information, Personal navigation, WWW.

1 Introduction

Location-based services (LBS) have become popular during recent years due to
increasingly wide availability of GPS positioning in multimedia mobile phones. For
instance, according to Nokia’s own estimate more than half of their phones would
include GPS by 2010-2012. In case of lacking GPS, positioning can also be based on
cellular network or even on IP address for rough estimation. It is therefore expected
that location-based services are emerging very fast to our everyday life via mobile
phones and other consumer electronics.

Locations-based services such as YellowPages¹, Google Maps² and Nokia Ovi
Services³ are traditionally based on databases where all entries have been explicitly
georeferenced when stored in the database. An alternative approach has been outlined

¹ http://en.02.fi/yellow+pages/
² http:/maps.google.com/
³ http://www.ovi.com/services/
in [10] and [8] based on web search and using ad-hoc georeferencing of the web-pages. We denote this approach as location-based search engine and emphasize it has seemingly small but significant distinction from traditional location-based services.

The bottleneck of this approach is that only very few pages have explicit georeferencing in form of geotaging, using address field or by other means. On the other hand, it is rather common that web-pages include street or postal addresses as free (non-tagged) text. According to [15], most of relevant services (especially commercial ones) can be found in this way. The main problem however, is how to find valid address elements from the web-pages both reliably and efficiently.

In this paper, we propose a method for extracting street names based on street-name prefix tree and a gazetteer. A potentially relevant web-page (by its content) is first analyzed by extracting all potential street address elements. The hypothesized addresses are then validated by the gazetteer. The pages (or part of them) with validated address are attached by the exact location obtained from the gazetteer and a prototype solution can be found at the MOPSI Search website.

Extraction of the potential street-name portion of the address field in most languages is very regular. It usually ends to way, drive, road, or in Finnish language to a suffix such as -katu, -kuja, -tie. A simple heuristic, used earlier in [8], performs a search for regular expressions with predefined endings (suffixes). However, not all street-names follow the predefined pattern and street-names that have a different suffix, such as Neulavahe, would not be detected. We therefore process all strings from the web-page since it can be done at the same cost when parsing the document.

Another problem is that we might detect as an address a portion of text that is not an actual address, causing a false detection. We therefore validate all hypothesized addresses by a gazetteer and discard the false detections. Our gazetteer is a geocoded database that contains geographical coordinates attached to address strings. As a side-product, the validation process provides the geocoding, i.e. converts the given address to a pair of coordinates. The process of recognizing geographic context is referred to as geoparsing and the process of assigning geographic coordinates to an address is known as geocoding.

One way to detect addresses from free form text is to build a classifier and let it detect addresses from the web-pages as in [21]. However, customizing the classifier to other languages and countries takes a considerable work as new ground truth tagged text corpus must be created by hand. In our approach, no ground truth tagging is needed. The only things needed are a gazetteer and simple rules on how the street name appears in relation to other address fields. Efficient use of the gazetteer is possible because we know the user’s current location and its interest area consists only on those services that are close to him. Therefore, we can build fast access structure to that partial gazetteer.

Matching of the potential address strings can be done brute force by comparing each word in the document to the retrieved table of street-names. However, this can be rather inefficient if the database is large. We therefore use the prefix tree as a search structure, which is critical for the performance of the matching. A set of prefix trees is constructed from all street-names in a given municipality and the ones in the

---

4 http://cs.joensuu.fi/mopsi/