

DCbot: Exploring the Web as Value-added Service for Location-based Applications

Mihály Jakob Matthias Grossmann Nicola Höhle Daniela Nicklas
University of Stuttgart, Institute of Parallel and Distributed Systems
Universitätsstraße 38, 70569 Stuttgart, Germany
[jakobmy | grossmms | hoenlena | danickla]@informatik.uni-stuttgart.de

1. Introduction

Location-based services (LBS) are typically mobile applications that adapt their behavior to the spatial context of the user, e.g. by providing maps and navigational information of the user's current position. Existing location-based applications rely on spatial data that is gathered and preprocessed especially for them and that is stored by particular data providers.

There is another large information space, the World Wide Web. Location-based applications can benefit from this new and additional information source, if, in a preprocessing step, web pages are mapped to locations. A model for this are Virtual Information Towers (VIT) [1], spatial web portals with a location and a visibility area that represents the region where the information is relevant. When a mobile user enters that area, the VIT becomes visible on his map. So far, we created VITs manually, but we want to automate this process.

Google Local Search [2] does already some location-based searching for web pages. However, the pages seem to be only mapped by street addresses. We want also to exploit other spatial information on web pages, to find pages which are relevant not only to a single building but also to regions or cities.

The approach presented in [5] exploits two different kinds of information: Names of locations like cities and states contained within a web page and the structure of links between web pages. However, we assume that the results are not precise enough for mobile users.

2. The DCbot

In Figure 1 we depict the overall scenario. DCbot processes HTML pages in the WWW like a crawler of a search engine. It analyses the pages using pre-defined rules and spatial knowledge and maps them to locations. These locations are not only points in space but regions: a

page could be related to a room in a building, to a single building or to a whole city. The result is a cartography of web pages.

DCbot (Figure 2) starts with initial web pages, follows all hyper links, and stores them in the URL Stack. The Content Handler analyses each web page referenced on the URL Stack. Finally, the cartography of web pages is stored in a Spatial Model Server.

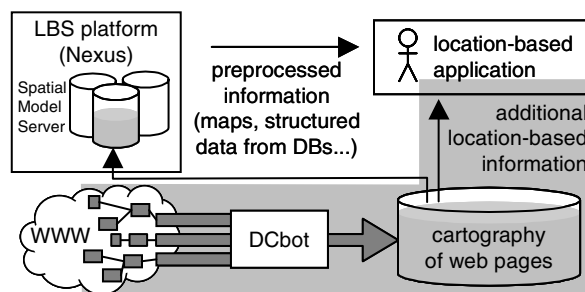


Figure 1. Overall scenario

The determination of the best geographic reference for a web page is a three step process. First, possible geographic references on the web page are identified by the analysis methods described in Section 2.2. Secondly, geographic reference candidates are cross-checked with the robot's geographic database GeoBase. Finally, confirmed geographic references are evaluated and for each web page the best reference is selected. The selection process is based on a weight matrix considering the accuracy of a spatial reference and the part of the web page the reference was extracted from. Afterwards, the page is stored in the Spatial Model Server. Web page entries of common locations are bundled into VITs.

2.1. Analyzed page parts

An early version of DCbot concentrated on the analysis of web page meta-tags that explicitly specified geographical information, e.g. the DC.Coverage tag from the Dublin Core Metadata Element Set [3]. Unfortunately, results of

several test runs have shown [4], that only 0.07% of web pages actually provide an explicit geographic reference. Therefore we chose to analyze the following page parts:

Hyperlink text. The text of hyperlinks are analyzed and assigned to the web page they are pointing to.

URL. Sometimes domain names give a hint about the content of a web site, for that reason they are analyzed.

Title. The title of a web page is one of the most important places to look for geographical references. In most cases it reflects the content of the web page.

Meta-tags. Meta-tags that explicitly provide geographical information are very rarely used. Nevertheless, common tags such as the keywords and description tags, which can provide valuable information about the page, are analyzed.

Body. The text content of a web page usually contains a significant number of words and sentences. The high amount of data requires sophisticated analysis methods.

2.2. Methods of analysis

The first step of calculating the best geographic reference on a web page is the extraction of geographic reference candidates. DCbot is currently set up for the analysis of German web pages. Language dependent parts stored in the keyword storage (Figure 2), can be replaced allowing the robot to process web pages in other languages.

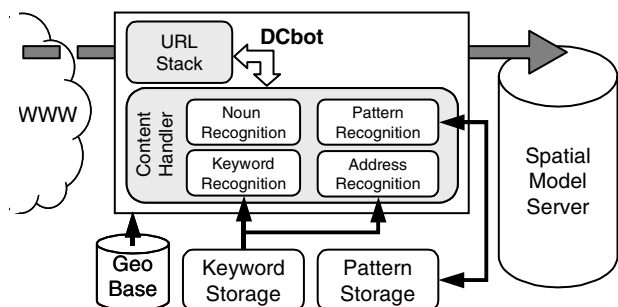


Figure 2. DCbot architecture

It is unlikely to find geographic coordinates on a web page. Therefore, DCbot searches for proper names of locations, e.g. names of institutions, important landmarks or postal addresses. DCbot uses the following candidate extraction methods:

Noun recognition. This method finds one word proper names. Depending on the size of a page part, all nouns, the first n nouns or the most frequent nouns are extracted.

Keyword recognition. This method identifies geographic reference candidates composed of several words based on a set of keywords, which indicate a possible proper name, e.g. *bridge* or *museum*. If a keyword is detected, several geographic reference candidates are extracted from its lexical environment.

Pattern recognition. This method stores lexical context patterns for each successfully verified geographic reference in the pattern storage (Figure 2). E.g. if a proper name of a geographic location was found on a web page after the term *'near to'*, then one of the patterns extracted would be *'near to <geo_ref>'*. If the term *'near to'* is encountered in subsequent pages, the following expression is treated like a geographic reference candidate. Patterns are rated by frequency allowing the robot to use an improving set of patterns.

Address recognition. DCbot uses a list of keywords indicating postal addresses, e.g. *street* or *square*. Address components are cross-checked with the robots geographical database GeoBase leveraging the hierarchical structure of the database.

3. Experiences

For this experiment, the set of start pages consisted of 100 web pages retrieved by a conventional web search engine by searching for web pages containing the string 'Stuttgart' in their title. During this run, DCbot analyzed about 25,000 web pages. On 50.89% of the pages, DCbot found spatial information. Table 1 shows the fractions for each type of spatial information except for very imprecise types of information like countries.

Table 1. Fractions of location information

type of location information	fraction
city	75.45%
address (zip code)	7.92%
university	3.71%
address (street)	2.34%
lake, college, museum, theater, hotel	1.21%

The by far most frequent type of location information are city names. This kind of information is not very useful, as it is associated with a relatively large area of relevance. Addresses are also a frequent type of location information, they allow a very precise determination of a location.

4. References

- [1] A. Leonhardi, U. Kubach, K. Rothermel, "Virtual Information Towers—A metaphor for intuitive, location-aware information access in a mobile environment", *Proc. of the 3rd Intl. Symposium on Wearable Computers*, 1999.
- [2] "Google Local Search", <http://local.google.com>
- [3] "Dublin Core Metadata Initiative", <http://dublincore.org>
- [4] M. Sütö, "Ortsbasierter Web-Zugriff", *Diploma thesis, University of Stuttgart*, 2002. (in German)
- [5] J. Ding, L. Gravano, N. Shivakumar, "Computing Geographical Scopes of Web Resources", *Proc. of the 26th Intl. Conference on Very Large Data Bases*, 2000.