

## Ontology Based Domain Specific Search of Crowdsourced OpenStreetMap Dataset and Wiki

### Authors

- **Stefan Keller**, University of Applied Sciences Rapperswil (HSR),  
*Switzerland*
- **Michel Ott**, University of Applied Sciences Rapperswil (HSR),  
*Switzerland*

**KEYWORDS :** information retrieval, ontology, text processing, open data, crowdsourcing

Finding relevant tags in the OpenStreetMap (OSM) database is a difficult endeavor even for experienced users of this still growing, well-known crowdsourced mapping project.

Within the OSM project there exists no authoritative and often no definition whatsoever of what a tag means. Tags are equivalent to attributes or key-value-pairs and serve also as map keys in interactive maps. OSM follows a decentralized yet collaborative manner with no central authorship of OSM. The advantage of this approach is the widespread participation in authoring content [2]. But the heterogeneity and the lack of structure makes it hard to satisfy the need to get a relevant answer while seeking for a tag or tag-scheme based on few simple real-world nouns expressed in local language.

With the project ‘TagFinder’ we are trying to satisfy the information-hunting task to find relevant tags based on a simple text query. This is the first and only web application which tries to deliver such functionality. For example, given a search term “Restaurant” a result page is being presented

with “amenity=restaurant” as the top ranked entry. If you search the german words “Autobahn” (highway) or “Gemeindehaus” (parish hall) in you won’t find the corresponding tags in the top 10 list of the result page of OSM Wiki. And in Taginfo service of OSM [4] there are no results found whatsoever.

In order to achieve this goal, mainly the OSM Wiki is being processed as input source as well statistics about the OSM database.

Within this contribution we describe the approach we have chosen in order to resolve this domain specific information retrieval task (Figure 1).

## Result

Search String Analysis	
<b>Search String:</b>	Restaurant
<b>Detected Language:</b>	de
<b>Preferred Value:</b>	restaurant

**FIGURE 1**

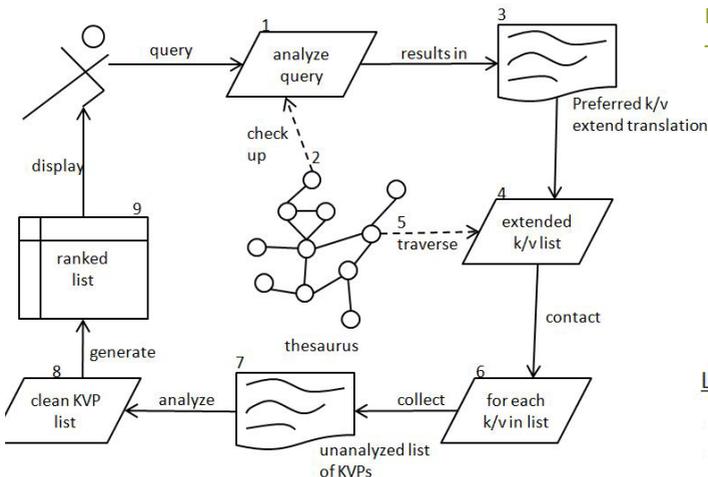
Information flow control of a user query within Tagfinder

Count	Key	Value	Filter
247604	amenity	restaurant	[+]
184	building	restaurant	[+]
158	bak:fac_type2	RESTAURANT	[+]
134	wpt_symbol	Restaurant	[+]
124	kvl_hro:amenity	restaurant	[+]
100	designation	Restaurant	[+]
90	sym	Restaurant	[+]
69	amenity	parking;restaurant;fuel	[+]
69	name:en	Bada Fish Restaurant	[+]

TagFinder has been implemented within a web application which offers industry-standard compliant webservice for Point-of-Interests from OSM [1] and is available for testing purposes (see [4]).

Given the running web application the user enters a query. First, the language of the input search term is determined by a thesaurus as a part of a domain-specific ontology and translated into an English equivalent. Thus, it is basically not important, in which language the end user enters the query. The thesaurus, which was developed especially for the OSM nomenclature, is using the Simple Knowledge Organization System (SKOS) schema [3]. SKOS is very suitable for the development of word lists, links, and their translations. In SKOS, there are two basic types of terms. On the one hand, the «concept scheme», which serves as a high-level group element, and the «Collection» for the collection of semantically related terms. The «concept scheme» is a defined controlled vocabulary such as the «Eating place», which subsumes all the possible terms of Restaurant. On the other hand there is the «concept», which is the smallest element in a SKOS thesaurus. A «Concept» describes a single term, including all its dependencies and translations. Thus, for example, concepts have father-son relationship or a reference to a preferred term.

With the help of the English equivalent, multiple requests are made on the Taginfo service [4] to find out which key-value pairs most likely corresponds to the term the user entered. Then the list will be filtered in order to eliminate duplicates. Then it is sorted in descending order of frequency and output (Figure 2).



**FIGURE 2**  
TagFinder result page

TagFinder offers also an application programming interface (API) so that it can be used by other applications like JOSM, the OSM editor.

Future work of TagFinder includes an extension of the database to cover the whole world as well as more sophisticated preprocessing and matching functions. There are several enhancements in consideration, like the management of the thesaurus. Here TagFinder is based on crowdsourcing principles itself. It uses special entries (a kind of templates) from OSM Wiki in order to indicate related terms. Then, in the preprocessing phase, we hope to enhance the result quality by filtering out proper names (like 'Windsor Castle'). The ranking can be further tuned depending on the fact, if the search term occurs in the key or value part of a tag candidate, and if there exists a Wiki page for the tag candidate. Finally, if there are too few results, fuzzy linguistic string matching functions, like trigrams, can support the retrieval of relevant tags given simple search terms.

---

[1] AUER, M., AND ZIPF, A. OpenStreetMap meets Open Geospatial Consortium - Zum Anwendungspotential einer Geodateninfrastruktur mit freiwillig gesammelten Geodaten. In Schilcher, M. (Hrsg.): Geoinformationssysteme. Beiträge zum 16. Münchner Fortbildungsseminar, 2011, pp. 106-113.

[2] CHAKRABARTI, S. Mining the Web - Discovering Knowledge from Hypertext Data. Morgan-Kaufmann Publishers, 352 p. (2002). ISBN 1-55860-754-4.

[3] SKOS, 2004. SKOS Simple Knowledge Organization System. Retrieved July 15, 2012 from <http://www.w3.org/2004/02/skos/>

[4] TAGFINDER, 2012. OpenStreetMap TagFinder Documentation. Retrieved July 15, 2012 from <http://giswiki.hsr.ch/TagFinder>

[5] TOPF, J. 2012. Taginfo API Documentation. Retrieved July 15, 2012 from <http://taginfo.openstreetmap.org/taginfo/apidoc>